



NONLINEAR ADAPTIVE EQUALIZATION WITH DATA SELECTION IN  
VLC SYSTEMS

Felipe Barboza da Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Wallace Alves Martins

Rio de Janeiro  
Março de 2018

NONLINEAR ADAPTIVE EQUALIZATION WITH DATA SELECTION IN  
VLC SYSTEMS

Felipe Barboza da Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

---

Prof. Wallace Alves Martins, D.Sc.

---

Prof. Paulo Sergio Ramirez Diniz, Ph.D.

---

Prof. Vinicius Nunes Henrique Silva, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2018

da Silva, Felipe Barboza

Nonlinear Adaptive Equalization with Data Selection in VLC Systems/Felipe Barboza da Silva. – Rio de Janeiro: UFRJ/COPPE, 2018.

XXIII, 110 p.: il.; 29, 7cm.

Orientador: Wallace Alves Martins

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2018.

Referências Bibliográficas: p. 99 – 110.

1. visible light communication. 2. data-selective algorithms. 3. set-membership filtering. 4. Volterra series. I. Martins, Wallace Alves. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

# Agradecimentos

Agradeço aos meus pais e irmão pelo suporte dado para que meu mestrado pudesse virar realidade. Agradeço em especial ao meu avô Oscar pela força que me tem dado desde o início da minha graduação.

Agradeço à minha namorada Jeniffer por me aguentar, e por estar comigo durante boa parte do tempo da escrita deste trabalho.

Ao meu orientador Wallace por sempre me exigir bastante, de forma que nossos trabalhos sempre resultem no melhor que nós podemos fazer, e pela dedicação incrível que demonstrou desde minha iniciação científica. Sou eternamente grato por tê-lo como orientador e amigo. Agradeço também aos outros professores do SMT, em especial Luiz Wagner, Marcello Campos e Diniz, os quais também considero meus amigos. Sou grato também ao Tadeu pela amizade e ao Markus, que além de ser meu amigo, em conjunto com Diniz, me ensinaram os fundamentos das principais contribuições deste trabalho. Agradeço também aos professores que tive aula durante as matérias do mestrado e doutorado, os quais, em sua maioria, se dedicaram ao máximo para passar o que sabiam e contribuíram à minha formação na área de pesquisa.

Aos meus amigos, agradeço especialmente a Matheus, Igor, Lucas Cinelli, Rafael, Gabriel, Wesley, Fonini, Curicica, Luiz e a todos os outros membros do grupo *Gaming Processing*. Aos que esqueci de mencionar, me perdoem.

Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) por me fornecerem suporte financeiro durante meu mestrado.

Agradeço aos prof. Paulo S. R. Diniz, prof. Vinicius N. H. Silva e prof. Wallace A. Martins por participarem da minha banca de mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## EQUALIZAÇÃO NÃO LINEAR ADAPTATIVA COM SELEÇÃO DE DADOS EM SISTEMAS VLC

Felipe Barboza da Silva

Março/2018

Orientador: Wallace Alves Martins

Programa: Engenharia Elétrica

Este trabalho propõe o uso de equalizadores com seleção de dados para comunicação por luz visível (VLC). Em tais sistemas, uma das principais fontes de não-linearidade é o diodo emissor de luz. Técnicas não-lineares tais qual a série de Volterra são empregadas para lidar com esse efeito. Contudo, a grande quantidade de parâmetros subjacentes ao modelo aumenta consideravelmente a complexidade computacional. Este trabalho propõe o uso de esquemas adaptativos com seleção de dados em série de Volterra. Trabalhando apenas com os dados de entrada que trazem inovação para o sistema, e assim evitando atualizações desnecessárias, é possível reduzir drasticamente a carga computacional associada ao uso da série de Volterra, de forma a proporcionar uma equalização não-linear.

Com o objetivo de reduzir ainda mais a carga computacional, este trabalho também propõe a combinação de diferentes algoritmos com seleção de dados. Assim, explorando a estrutura da série de Volterra, é possível empregar um filtro para cada um de seus *kernels*. Cada um desses filtros atualiza mais elementos conforme a ordem do *kernel* aumenta. Portanto, atribuindo limiares de erro maiores para esses filtros, é possível diminuir suas taxas de atualização, assim reduzindo a complexidade computacional total. É também proposto o uso de algoritmos semi-cegos e com seleção de dados para diminuir o montante de dados usados para treinar os filtros adaptativos, enquanto reduz a complexidade computacional.

Uma plataforma computacional para simulação de sistemas VLC também é uma das propostas deste trabalho. Os componentes-chave deste sistema são apresentados, cujos modelos são baseados na descrição de suas propriedades físicas feitas neste trabalho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## NONLINEAR ADAPTIVE EQUALIZATION WITH DATA SELECTION IN VLC SYSTEMS

Felipe Barboza da Silva

March/2018

Advisor: Wallace Alves Martins

Department: Electrical Engineering

This work proposes the use of data-selective equalizers for visible light communication (VLC). In such systems, one of the main sources of nonlinearity is the light-emitting diode. Nonlinear techniques such as Volterra series are employed to cope with this effect. However, the large number of parameters underlying to the model yielded by such series grows considerably the computational complexity. This work proposes the use of data-selective adaptive schemes in Volterra series. By working only with input data that brings novelty to the system, thus avoiding unnecessary parameter updates, one can reduce drastically the high computational burden associated with the use of Volterra series, therefore, providing an efficient nonlinear equalizer.

In order to reduce even more the computational burden, this work also proposes the combination of different data-selective algorithms. Thus, by exploring the structure of the Volterra series, one can employ a filter for each of its kernels. Each of these filters update more elements as the order of the kernel increases. Therefore, by assigning larger error thresholds to these filters, it is possible to decrease their update rates, thus reducing the overall computational complexity. Also, it is proposed the use of data-selective semi-blind algorithms to decrease the amount of data used to train the adaptive filters, while reducing the computational complexity.

A computational platform for simulation of VLC systems is also one of the proposals of this work. The key components of this system are presented, whose models are based on the description of their physical properties performed in this work.

# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xiv</b>
<b>List of Symbols</b>	<b>xvi</b>
<b>List of Abbreviations</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 VLC System Modulations . . . . .	2
1.2 Historical Development . . . . .	3
1.3 VLC Main Challenges . . . . .	6
1.4 Contributions of this Work . . . . .	6
1.5 Organization of the Text . . . . .	8
1.6 Publications . . . . .	9
<b>2 VLC Main Components</b>	<b>10</b>
2.1 VLC System Overview . . . . .	10
2.2 Light-Emitting Diode . . . . .	11
2.3 Free-Space Optical Channel . . . . .	14
2.4 Photodiode . . . . .	15
<b>3 Data-Selective Volterra Adaptive Filters</b>	<b>19</b>
3.1 Adaptive Filtering . . . . .	19
3.2 Linear-in-input SMAF . . . . .	21
3.2.1 SM-PNLMS . . . . .	21
3.2.2 Modified BEACON . . . . .	22
3.3 Volterra-Based SMAF . . . . .	24
3.4 Performance Evaluation . . . . .	27
3.4.1 Simulation Methodology and Figures of Merit . . . . .	28
3.4.2 Simulation Setup . . . . .	28
3.4.3 Results for System Identification . . . . .	29

3.4.4	Results for Channel Equalization . . . . .	30
<b>4</b>	<b>Semi-Blind Data-Selective and <math>C</math>-Threshold Volterra Adaptive Filtering</b>	<b>34</b>
4.1	Semi-Blind Volterra-based SMAF . . . . .	35
4.1.1	VSBSM-PNLMS . . . . .	35
4.1.2	VSB Modified BEACON . . . . .	36
4.1.3	From Supervised to Blind Training . . . . .	37
4.2	$C$ -Threshold Volterra-based SMAF . . . . .	38
4.2.1	Computational Efficiency of Volterra Double-Threshold Algorithm . . . . .	42
4.3	Semi-blind $C$ -Threshold Volterra-based SMAF . . . . .	44
4.4	Performance Evaluation . . . . .	44
4.4.1	Simulation Methodology and Figures of Merit . . . . .	45
4.4.2	Simulation Setup . . . . .	45
4.4.3	Results for System Identification . . . . .	46
4.4.4	Equalization Results for VDTSM-PNLMS Algorithm . . . . .	47
4.4.5	Results for VSBSM-PNLMS and VSBM-BEACON Algorithms . . . . .	51
4.4.6	Results for VSBDTSM-PNLMS Algorithms . . . . .	53
4.4.7	Conclusions of VSBDTSM-PNLMS Simulation . . . . .	53
4.4.8	Final Remarks . . . . .	57
<b>5</b>	<b>Modeling the Key Components of a VLC System</b>	<b>58</b>
5.1	LED Model . . . . .	59
5.1.1	Frequency-Response Model . . . . .	60
5.1.2	$I$ - $V$ Curve Model . . . . .	60
5.1.3	Electrical-to-Optical Conversion Model . . . . .	63
5.1.4	Optical Characteristics . . . . .	64
5.2	Free-Space Optical Channel Model . . . . .	64
5.3	Receiver Model . . . . .	67
<b>6</b>	<b>Results and Discussions</b>	<b>69</b>
6.1	Performance Evaluation . . . . .	69
6.1.1	Simulation Methodology and Figures of Merit . . . . .	69
6.1.2	Simulation Setup . . . . .	70
6.1.3	Results of VSM-PNLMS and VM-BEACON Algorithms . . . . .	71
6.1.4	Conclusions of VSM-PNLMS and VM-BEACON Simulations . . . . .	72
6.1.5	Results of VDTSM-PNLMS Algorithm . . . . .	73
6.1.6	Conclusions of VDTSM-PNLMS Simulations . . . . .	73
6.1.7	Results of VSBSM-PNLMS and VSBM-BEACON Algorithms . . . . .	76

6.1.8	Conclusions of VSBSM-PNLMS and VSBM-BEACON Simulations . . . . .	77
6.1.9	Results of VSBDTSM-PNLMS Algorithm . . . . .	78
6.1.10	Conclusions of VSBDTSM-PNLMS Simulations . . . . .	79
6.1.11	Final Remarks . . . . .	82
<b>7</b>	<b>Conclusions and Future Works</b>	<b>83</b>
7.1	Concluding Remarks . . . . .	83
7.2	Research Directions . . . . .	84
<b>A</b>	<b>Derivation of the Adaptive Algorithms</b>	<b>85</b>
A.1	SM-PNLMS . . . . .	85
A.2	BEACON . . . . .	86
A.3	M-BEACON . . . . .	90
A.4	Volterra Semi-blind SM-PNLMS . . . . .	91
<b>B</b>	<b>Pseudo-codes of the Adaptive Algorithms</b>	<b>94</b>
B.1	SM-PNLMS . . . . .	94
B.2	M-BEACON . . . . .	95
B.3	VSBSM-PNLMS . . . . .	95
B.4	VSBM-BEACON . . . . .	96
B.5	VDTSM-PNLMS . . . . .	97
B.6	VSBDTSM-PNLMS . . . . .	98
	<b>Bibliography</b>	<b>99</b>

# List of Figures

1.1	Illustration of the optical telegraph. Reprinted from [1]. . . . .	4
1.2	Illustration of the photophone transmitter. Reprinted from [2]. . . . .	4
1.3	Illustration of the photophone receiver. Reprinted from [3]. . . . .	5
1.4	Illustration of PLC and VLC systems in an indoor environment. . . . .	5
2.1	Overview of a VLC system. . . . .	11
2.2	Typical LED $I$ - $V$ curve. . . . .	13
2.3	Illustration of the depletion region of a $pn$ junction. . . . .	13
2.4	Sketches of the spectral response (in terms of wavelength) of LEDs employed in white color-based communications. . . . .	14
2.5	Photodiodes' modes of operation. . . . .	16
2.6	Characteristic $I$ - $V$ curve in reverse-bias mode for different values of incident power. . . . .	17
2.7	Typical shape of spectral response of photodiodes. . . . .	18
3.1	General configuration of an adaptive filter. . . . .	20
3.2	System identification scheme. . . . .	20
3.3	Channel equalization scheme. . . . .	20
3.4	Adaptive Volterra decision-feedback equalizer. . . . .	27
3.5	MSE (a) and misalignment (b) for nonlinear system identification. . . . .	29
3.6	MSE for different memory sizes in the feedforward equalization case. . . . .	30
3.7	BER for different memory sizes in the feedforward equalization case. . . . .	31
3.8	Steady-state MSE for different memory sizes for VSM-PNLMS. . . . .	32
3.9	Steady-state MSE for different memory sizes for VM-BEACON. . . . .	32
3.10	MSE (a) and BER (b) for $M_{FF} = 4$ and $M_{FB} = 0$ in the DFE case. . . . .	33
4.1	4-PAM decision regions and the subregion defined by $\eta$ . . . . .	38
4.2	Generic topology of the combination of $C$ adaptive filters. . . . .	41
4.3	MSE (a) and misalignment (b) for VDTSM-PNLMS algorithm. . . . .	47
4.4	NUE for VDTSM-PNLMS algorithm using different values of $\bar{\gamma}_2$ . The red line indicates the number of updated elements in the VSM-PNLMS case. . . . .	48

4.5	MSE for different memory sizes in the feedforward equalization case for VDTSM-PNLMS algorithm. . . . .	48
4.6	BER for different memory sizes in the feedforward equalization case for VDTSM-PNLMS algorithm. . . . .	49
4.7	NUE for VDTSM-PNLMS algorithm in the feedforward equalization case using different values of $\bar{\gamma}_2$ . The blue and red lines represent the NUE in VSM-PNLMS case for $M_{FF} = 3$ and $M_{FF} = 3$ , respectively. . . . .	49
4.8	MSE (a) and BER (b) for $M_{FF} = 4$ and $M_{FB} = 0$ using different thresholds $\bar{\gamma}_2$ in the DFE case for VDTSM-PNLMS. . . . .	50
4.9	NUE for VDTSM-PNLMS algorithm in the DFE equalization case using different values of $\bar{\gamma}_2$ . The red line represents the NUE in the VSM-PNLMS case. . . . .	50
4.10	MSE (a) and BER (b) for $M_{FF} = 8$ using different thresholds $\bar{\gamma}_2$ in the feedforward case for VSBSM-PNLMS and VSBM-BEACON. . . . .	52
4.11	MSE (a) and BER (b) for $M_{FF} = 8$ using different thresholds $\bar{\gamma}_2$ in the DFE case for VSBSM-PNLMS and VSBM-BEACON. . . . .	52
4.12	Steady-state MSE for different values of $\bar{\gamma}_2$ and $\eta = 0.1$ in the feedforward (a) and DFE (b) cases for VSBDTSM-PNLMS. . . . .	54
4.13	MSE (a) and BER (b) for $M_{FF} = 8$ , $\eta = 0.1$ , and using different thresholds $\bar{\gamma}_2$ in the feedforward case for VSBDTSM-PNLMS. . . . .	54
4.14	NUE for VSBDTSM-PNLMS algorithm in the feedforward equalization case using different values of $\bar{\gamma}_2$ . The red line indicates the NUE in the VSBSM-PNLMS case. . . . .	55
4.15	MSE (a) and BER (b) for $M_{FF} = 8$ , $\eta = 0.1$ , and using different thresholds $\bar{\gamma}_2$ in the DFE case for VSBDTSM-PNLMS. . . . .	55
4.16	NUE for VSBDTSM-PNLMS algorithm in the DFE equalization case using different values of $\bar{\gamma}_2$ . The red line indicates the NUE in the VSBSM-PNLMS case. . . . .	56
5.1	Simplified VLC model. . . . .	59
5.2	LED's modes of operation. . . . .	61
5.3	Equivalent model (a) and (b) $I$ - $V$ curve for ideal approximation. . . . .	61
5.4	(a) Equivalent model and (b) $I$ - $V$ curve for piecewise approximation. . . . .	61
5.5	$I$ - $V$ curve for exponential approximation. . . . .	62
5.6	Nonlinear behavior of luminous intensity for different $k$ 's. . . . .	63
5.7	Complete LED model. . . . .	64
5.8	Shape of generalized Lambertian radiation pattern for different half-power angles. . . . .	65
5.9	Geometry of the LED and PD. . . . .	65

5.10	$I$ - $V$ curve in reverse-bias mode for different values of incident power.	68
6.1	MSE (a) and BER (b) in the feedforward case for VSM-PNLMS and VM-BEACON.	71
6.2	MSE (a) and BER (b) for $M_{\text{FF}} = 8$ in the DFE case for VSM-PNLMS and VM-BEACON.	72
6.3	MSE for different values of $\bar{\gamma}_2$ and MI in feedforward case for VDTSM-PNLMS.	73
6.4	BER for different values of $\bar{\gamma}_2$ and MI in feedforward case for VDTSM-PNLMS.	74
6.5	NUE for different values of $\bar{\gamma}_2$ and MI in feedforward case for VDTSM-PNLMS. The red line indicates the NUE in the VSM-PNLMS case.	74
6.6	MSE for different values of $\bar{\gamma}_2$ and MI in DFE case for VDTSM-PNLMS.	74
6.7	BER for different values of $\bar{\gamma}_2$ and MI in DFE case for VDTSM-PNLMS.	75
6.8	NUE for different values of $\bar{\gamma}_2$ and MI in DFE case for VDTSM-PNLMS. The red line indicates the NUE in the VSM-PNLMS case.	75
6.9	MSE for different values of MI in feedforward case for VSBSM-PNLMS and VSBM-BEACON.	76
6.10	BER for different values of MI in feedforward case for VSBSM-PNLMS and VSBM-BEACON.	77
6.11	MSE for different values of MI in DFE case for VSBSM-PNLMS and VSBM-BEACON.	77
6.12	BER for different values of MI in DFE case for VSBSM-PNLMS and VSBM-BEACON.	78
6.13	Steady-state MSE for different values of $\bar{\gamma}_2$ and $\eta = 0.1$ in the feedforward case for VSBDTSM-PNLMS.	79
6.14	Steady-state MSE for different values of $\bar{\gamma}_2$ and $\eta = 0.1$ in the DFE case for VSBDTSM-PNLMS.	79
6.15	MSE for different thresholds $\bar{\gamma}_2$ and MIs in the feedforward case for VSBDTSM-PNLMS.	80
6.16	BER for different thresholds $\bar{\gamma}_2$ and MIs in the feedforward case for VSBDTSM-PNLMS.	80
6.17	NUE for different thresholds $\bar{\gamma}_2$ and MIs in the feedforward case for VSBDTSM-PNLMS. The red line indicates the NUE in the VSBSM-PNLMS case.	81
6.18	MSE for different thresholds $\bar{\gamma}_2$ and MIs in the DFE case for VSBDTSM-PNLMS.	81
6.19	BER for different thresholds $\bar{\gamma}_2$ and MIs in the DFE case for VSBDTSM-PNLMS.	81

6.20	NUE for different thresholds $\bar{\gamma}_2$ and MI in the DFE case for VSBDTSM-PNLMS. The red line indicates the NUE in the VSBSM-PNLMS case . . . . .	82
------	--	----

# List of Tables

3.1	Iterations until steady state / Average update rates for feedforward equalization. . . . .	31
4.1	Description of parameters and figures of merit. . . . .	46
4.2	Iterations until convergence and update rates for VDTSM-PNLMS algorithm using different thresholds $\bar{\gamma}_2$ . . . . .	47
4.3	Iterations until convergence and update rates for $M_{FF} = 3/M_{FF} = 4$ . . . . .	49
4.4	Iterations until convergence and update rates for $M_{FF} = 4$ and $M_{FB} = 0$ . . . . .	50
4.5	Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the feedforward case. . . . .	51
4.6	Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the DFE case. . . . .	52
4.7	Average number of iterations until unsupervised period and update rates using $\eta = 0.1$ for VSBDTSM-PNLMS in the feedforward case. . . . .	54
4.8	Average number of iterations until unsupervised period and update rates using $\eta = 0.1$ for VSBDTSM-PNLMS in the DFE case. . . . .	56
6.1	Description of the proposed algorithms and sections in which they were presented. . . . .	70
6.2	Iterations until steady state / Average update rates using feedforward equalization for VSM-PNLMS and VM-BEACON. . . . .	71
6.3	Iterations until steady state / Average update rates using DFE equalization for VSM-PNLMS and VM-BEACON. . . . .	72
6.4	Iterations until steady state / Average update rates $\nu_1$ / Average update rates $\nu_2$ using feedforward equalization for VDTSM-PNLMS. . . . .	73
6.5	Iterations until steady state / Average update rates $\nu_1$ / Average update rates $\nu_2$ using DFE equalization for VDTSM-PNLMS. . . . .	75
6.6	Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the feedforward case. . . . .	76
6.7	Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the DFE case. . . . .	77

6.8	Average number of iterations until unsupervised period / Average update rates $\nu_1$ / Average update rates $\nu_2$ using $\eta = 0.1$ for VSBDTSM-PNLMS in the feedforward case. . . . .	80
6.9	Average number of iterations until unsupervised period / Average update rates $\nu_1$ / Average update rates $\nu_2$ using $\eta = 0.1$ for VSBDTSM-PNLMS in the DFE case. . . . .	82

# List of Symbols

$A$	Photodiode detection area, p. 59
$\alpha$	Voltage constant chosen to keep the LED input signal inside a predefined range, p. 58
$b$	Reference signal during unsupervised period, p. 36
$\beta$	Amplification gain induced by transimpedance amplifier, p. 11
$C$	Number of filters employed in the combination of filters scheme, p. 38
$c[k]$	Data signal from a given modulation scheme, p. 58
$D$	Delay in samples, p. 20
$d$	Desired or reference signal, p. 19
$\delta[k]$	Discrete unit impulse, p. 66
$D_s$	Delay spread of the channel, p. 66
$e$	Output error, p. 19
$e_b$	Output error during unsupervised period, p. 36
$e_c$	Output error of the $c^{\text{th}}$ adaptive filter, p. 39
$e_{c_b}$	Output error of the $c^{\text{th}}$ filter during unsupervised period, p. 44
$\mathbf{e}_{\text{SB}}$	Output error vector, p. 37
$\eta$	Threshold that controls the width of the sub-region area within the decision region, p. 38
$F(\omega)$	Continuous-time frequency response model for LED, p. 60
$F_s(e^{j\omega})$	Discrete-time version of $F(\omega)$ , p. 60

$\mathbf{G}$	Diagonal weighting matrix, p. 22
$\bar{\gamma}$	Error threshold employed in some data-selective algorithms, p. 21
$\bar{\gamma}_1$	Error threshold for the filter $\mathbf{w}_1$ , p. 42
$\bar{\gamma}_2$	Error threshold for the filter $\mathbf{w}_2$ , p. 42
$\bar{\gamma}_{\text{SB}}$	Error threshold employed to switch from supervised to unsupervised period, p. 37
$\mathbf{G}_c$	Diagonal weighting matrix of the $c^{\text{th}}$ adaptive filter, p. 40
$g[k]$	Zero-mean white Gaussian noise at photodiode, p. 59
$\mathcal{G}(\theta)$	Concentrator gain, p. 65
$\mathcal{H}$	<i>Constraint set</i> , p. 21
$H(0)$	DC gain of the optical channel, p. 65
$\Phi_{1/2}$	LED's half-power angle, p. 64
$h[k]$	Time-invariant filter that emulates the optical channel, p. 59
$H\{\cdot\}$	Optical channel operator, p. 11
$\mathcal{H}_{\text{Volterra}}$	<i>Constraint set</i> considering Volterra series, p. 25
$I(t)$	Continuous-time current of the LED, p. 62
$I[k]$	Discrete-time version of $I(t)$ , p. 62
$\mathcal{I}[k]$	Luminous intensity signal, p. 59
$I_p$	Resulting current at the photodiode, p. 67
$I_s$	Saturation current, p. 62
$k$	Knee factor that adjusts the smoothness of the luminous intensity saturation, p. 63
$\kappa$	Sparse information factor, p. 22
$\kappa_c$	Sparse information factor of the $c^{\text{th}}$ adaptive filter, p. 40
$K_B$	Boltzmann's constant, p. 62
$L$	Length of parameter vector, p. 19

$L_1$	Number of updated elements when $\mathbf{w}_1$ updates, p. 43
$\lambda$	Forgetting factor, p. 23
$L\{\cdot\}$	Electrical-to-optical conversion operator, p. 11
$L_{\text{SB}}$	Length of the output error vector, p. 37
$l(t)$	Continuous-time signal emitted by LED, p. 10
$\tilde{l}(t)$	Continuous-time signal $l(t)$ after optical channel operation, p. 11
$M$	Memory size of an adaptive filter, p. 24
$M_{\text{FB}}$	Memory size of feedback filter, p. 43
$M_{\text{FF}}$	Memory size of feedforward filter, p. 43
$\mu$	Step-size, p. 22
$\mu_c$	Step-size of the $c^{\text{th}}$ adaptive filter, p. 40
$N$	Volterra series order, p. 24
$n_{\text{LED}}$	Ideality factor of the LED, p. 62
$\nu$	Update rate of standard Volterra-based algorithms, p. 43
$\nu_1$	Update rate of $\mathbf{w}_1$ , p. 43
$\nu_2$	Update rate of $\mathbf{w}_2$ , p. 43
$\nu_{\text{BP}}$	Update rate of standard Volterra-based algorithms during unsupervised period, p. 45
$\nu_{\text{BP}_1}$	Update rate of $\mathbf{w}_1$ during unsupervised period, p. 45
$\nu_{\text{BP}_2}$	Update rate of $\mathbf{w}_2$ during unsupervised period, p. 45
$\nu_{\text{SP}}$	Update rate of standard Volterra-based algorithms during supervised period, p. 45
$\nu_{\text{SP}_1}$	Update rate of $\mathbf{w}_1$ during supervised period, p. 45
$\nu_{\text{SP}_2}$	Update rate of $\mathbf{w}_2$ during supervised period, p. 45
$\omega_s$	Sampling frequency, p. 60
$P_r$	Received power at photodiode, p. 67

$\Psi$	<i>Exact membership set</i> , p. 21
$q$	Electron's charge magnitude, p. 62
$R\{\cdot\}$	Optical-to-electrical conversion at the photodiode, p. 11
$r(t)$	Continuous-time signal received by photodiode, p. 10
$\rho$	Electrical-optical conversion factor, p. 63
$r[k]$	Discrete-time version of $r(t)$ , p. 59
$R_L$	Ratio between the $N^{\text{th}}$ -order Volterra and linear-in-input filter lengths, p. 25
$R$	Photodiode responsivity, p. 59
$\mathbf{S}$	Matrix that parametrizes an ellipsoid in BEACON algorithm, p. 22
$\sigma$	Convex combination parameter, p. 42
$\sigma_c$	Combination parameter, p. 39
$\tilde{\mathbf{S}}$	Matrix that parametrizes an ellipsoid in Modified BEACON algorithm, p. 24
$\Theta$	<i>Feasibility set</i> , p. 21
$\mathcal{T}(\theta)$	Optical filter gain, p. 65
$T$	Absolute temperature, p. 62
$\mathbf{U}$	Matrix that selects the elements of $\mathbf{w}$ that will be updated, p. 38
$\mathbf{U}_{\text{DFE}_c}$	Matrix that selects the elements of $\mathbf{w}_c$ that will be updated in the DFE case, p. 42
$\mathbf{U}_c$	Matrix that selects the elements of $\mathbf{w}_c$ that will be updated, p. 41
$\tilde{\mathbf{U}}$	Complementary matrix of $\mathbf{U}$ , p. 39
$u(t)$	Unit step function, p. 66
$V_{\text{DC}}$	DC voltage bias, p. 58
$V_{\text{in}}(t)$	Continuous-time signal at the input of the LED, p. 10

$V_{\text{in}}[k]$	Discrete-time version of $V_{\text{in}}(t)$ , p. 58
$V_{\text{max}}$	Maximum amplitude of LED input signal, p. 58
$V_p(\cdot)$	Operator that employs the $p^{\text{th}}$ Volterra series in a given input signal, p. 26
$V_q(\cdot)$	Operator that employs the $q^{\text{th}}$ Volterra series in a given input signal, p. 26
$V_T$	Thermal voltage of the LED, p. 62
$\mathbf{w}$	Parameter vector, p. 19
$\mathbf{w}_c$	Parameter vector of the $c^{\text{th}}$ filter, p. 38
$\mathbf{w}_{\text{FB}}$	Parameter vector of the feedback filter, p. 26
$\mathbf{w}_{\text{FF}}$	Parameter vector of the feedforward filter, p. 26
$\mathbf{x}$	Input signal vector, p. 19
$\mathbf{x}_N$	$N$ th-order Volterra input signal vector, p. 24
$\mathbf{x}_{\text{FB}_q}$	$q^{\text{th}}$ -order Volterra input signal vector of the feedback filter, p. 26
$\mathbf{x}_{\text{FF}_p}$	$p^{\text{th}}$ -order Volterra input signal vector of the feedforward filter, p. 26
$y$	Output of the adaptive filter, p. 19
$y_c$	Output of the $c^{\text{th}}$ adaptive filter, p. 39
$\tilde{y}$	Detected symbol in the DFE equalization case, p. 26
$\zeta$	Distance between LED and photodiode, p. 65

# List of Abbreviations

ACO-OFDM	Asymmetrically clipped orthogonal frequency-division multiplexing, p. 3
ADO-OFDM	Asymmetrically DC-biased orthogonal frequency-division multiplexing, p. 3
APD	Avalanche photodiode, p. 16
BEACON	Bounding ellipsoidal adaptive constrained least-squares, p. 7
BER	Bit error rate, p. 6
CMA	Constant modulus algorithm, p. 35
CSK	Color-shift keying, p. 2
DCO-OFDM	DC-biased orthogonal frequency-division multiplexing, p. 2
DD	Direct detection, p. 11
DFE	Decision-feedback equalization, p. 26
DMT	Discrete multitone, p. 4
FOV	Field of view, p. 65
FSK	Frequency-shift keying, p. 2
FSO	Free-space optical, p. 8
IDFT	Inverse discrete Fourier transform , p. 3
IM	Intensity modulation, p. 2
IOT	Internet-of-Things, p. 1
IR	Infra-red, p. 6
ISI	Intersymbol interference, p. 66

LD	Laser diode, p. 5
LED	Light-emitting diode, p. 1
LOS	Line-of-sight, p. 64
M-BEACON	Modified BEACON, p. 24
MIMO	Multiple-input multiple-output, p. 6
MI	Modulation index, p. 58
MSE	Mean square error, p. 9
NLOS	Non-line-of-sight, p. 66
NUE	Average number of updated elements, p. 43
OBE	Optimal bounding ellipsoid, p. 22
OFDM	Orthogonal frequency-division multiplexing, p. 2
OOK	On-off keying, p. 2
PAM	Pulse amplitude modulation, p. 2
PD	Photodiode, p. 10
PIN	P-Intrinsic-N diode, p. 4
PLC	Power line communication, p. 3
PPM	Pulse position modulation, p. 2
QAM	Quadrature amplitude modulation, p. 2
RF	Radiofrequency, p. 1
RGB	Red-Green-Blue, p. 4
RGBY	Red-Green-Blue-Yellow, p. 5
RKHS	Reproducing kernel Hilbert space, p. 25
RLI	Run length limited, p. 6
RLS	Recursive least-squares, p. 19
SM-PNLMS	Set-membership proportionate normalized least-mean-square, p. 7

SM-PUPNLMS	SM-partial-update-PNLMS, p. 39
SMAF	Set-membership adaptive filtering, p. 7
SNR	Signal-to-noise ratio, p. 14
TIA	Transimpedance amplifier, p. 10
V-PNLMS	Volterra PNLMS, p. 27
V-RLS	Volterra RLS, p. 27
VDTSM-PNLMS	Volterra double-threshold SM-PNLMS, p. 41
VLC	Visible light communication, p. 1
VM-BEACON	Volterra M-BEACON, p. 25
VSBDTSM-PNLMS	Volterra semi-blind double-threshold SM-PNLMS, p. 44
VSBSM-BEACON	Volterra semi-blind M-BEACON, p. 37
VSBSM-PNLMS	Volterra semi-blind SM-PNLMS, p. 35
VSM-PNLMS	Volterra SM-PNLMS, p. 25
WDM	Wavelength division multiplexing, p. 4
pc-LED	Phosphor-converted LED, p. 13

# Chapter 1

## Introduction

Visible light communication (VLC) is a technique that employs visible light to transmit data. When compared to traditional radiofrequency (RF) communications, the key components that enable VLC to work are a light-emitting diode (LED), responsible for transforming electrical signals into light signals at the transmitter, and a photodiode, responsible for converting this optical signal into a corresponding current level at the receiver end. VLC can be employed in a large range of applications, such as: short range communication systems, working as a complement to or even a substitute for RF systems [4, 5]; intelligent transport systems, providing communications among vehicles [6, 7]; in the context of Internet-of-Things (IOT), where toys are communicating to each other using LEDs [8]; and indoor localization systems [9, 10]. Considering these applications, VLC systems feature some advantages if compared to its RF counterparts due to the following factors:

- Increase in the capacity: VLC provides an unregulated spectrum from 400 THz to 800 THz, while RF-based communications occupy a band from 3 kHz to 300 GHz, which means VLC overall frequency band is 10,000 higher than RF's. This is one of the reasons why VLC is considered to be an alternative solution that addresses the RF spectrum scarcity by relieving this crowded frequency band. In addition, mobile data traffic is expected to increase 7 times from 2016–2021 [11]. Thereby, mobile operators will focus on public Wi-Fi hotspots and other alternative techniques.
- Non interference: VLC signals do not interfere with RF waves [12]. Therefore, this technology can be used as a complementary system, in addition to allowing for its application in hospitals and airplanes.
- Security and reuse: Unlike RF waves, which penetrate walls, visible light cannot pass through opaque objects. Thus, once data is converted into light, one can hardly eavesdrop it. Moreover, the VLC transmission is highly directional,

which means that transmitter and receiver should have a considerable alignment in most cases. While this is usually regarded as a drawback, one can benefit from it since a lot of different systems can operate within the same indoor environment with mild interference.

- **Low cost of deployment:** LEDs have been replacing other sources of light as incandescent and fluorescent bulbs. Besides, it is expected that approximately 85% of all illumination in the United States will be provided by LEDs until 2035 [13]. VLC can benefit from this pre-existent infrastructure, and, with the addition of few cheap front-end devices [14], can be deployed at low cost. In addition, the key components of a VLC system are inexpensive if compared to their RF counterparts.
- **Energy efficiency:** LEDs are considered as a green technology. Their luminous efficacy can vary from 68 up to nearly 140 lumens/watt, while fluorescent and incandescent lamps present at maximum 70 and 15 lumens/watt, respectively [15]. Supposing that all sources of light are replaced by LEDs, this could yield a reduction in the world energy consumption of 50% [16].
- **Not hazardous to healthy:** Unlike RF systems, there are no health regulations to restrict the transmitted power.

## 1.1 VLC System Modulations

VLC systems transmit data through intensity modulation (IM), where the data to be transmitted is modulated varying the luminous intensity of the emitted light [17]. Unlike RF waves, the light is an incoherent source, which hinders the use of heterodyne receivers [17]. Hence, typical RF modulations such as frequency-shift keying (FSK) cannot be employed. Instead, VLC systems are based on other modulations, e.g., pulse amplitude modulation (PAM), on-off keying (OOK), pulse-position modulation (PPM), color-shift keying (CSK), spatial modulation, among others. In addition, the signal transmitted by the LED is strictly positive and real, which means that a certain DC level must be added to the symbols before transmission. Therefore, some modulations such as quadrature amplitude modulation (QAM), and transmission schemes like orthogonal frequency-division multiplexing (OFDM), should be adapted to work in a VLC system. For instance, a system may employ one LED to transmit the real symbols, while the other transmits the imaginary ones.

Regarding OFDM schemes, there are works in the literature addressing some adaptations to deal with the aforementioned issues. For example, the DC-biased optical OFDM (DCO-OFDM) adds a DC value to the symbols to ensure their non-

negativeness, besides making use of symmetry properties of the inverse discrete Fourier transform (IDFT) to map the complex symbols into real symbols [18–20]. Instead of adding the DC level, which may decrease the efficiency in terms of average optical power [18], the asymmetrically clipped optical OFDM (ACO-OFDM) modulates only the odd subcarriers, which turns the signal after the IDFT operation antisymmetric. Hence, the same information is contained in both positive and negative elements of the signal [18, 21, 22]. Nonetheless, by transmitting useful data only in the odd subcarriers, the spectral efficiency of the system is reduced [20]. In order to enhance the average optical power and spectral efficiency, the authors in [23] proposed the ADO-OFDM, which employs ACO-OFDM on the odd subcarriers and DCO-OFDM on the even subcarriers.

After describing the motivation for using VLC systems, as well as the most popular modulations and transmission schemes, the historical development of such technology will be presented.

## 1.2 Historical Development

The most popular device to transmit data using visual signals is the optical telegraph. It was invented in 1792 by Claude Chappe and became very popular from late eighteenth century until early nineteenth century in Europe [24]. The optical telegraph is composed by 2 long arms, which are connected by an horizontal bar, as illustrated in Figure 1.1. These arms could display seven positions each, while the horizontal bar could perform four different inclination angles, totaling  $7 \times 7 \times 4 = 196$  distinct possible symbols.

In 1880, the distinguished inventor Alexander Graham Bell created the first device to transmit voice by means of light, named photophone, whose working principle is described as follows: the transmitter modulates a light beam by using mirrors, which under the action of voice, becomes more convex or concave, scattering or condensing the light beam, as shown in Figure 1.2. Figure 1.3 illustrates the receiving process, which is based on the conversion of the incident beam of light into variations of air pressure by a selenium cell positioned at the focus of a mirror [25].

More recently, VLC technology flourished in the late 1990s in Japan, when LED traffic lights were employed to broadcast audio messages to drivers [6]. In the early 2000s, as white LEDs began to be employed for indoor illumination, research pioneers from Keio University, named Yuichi Tanaka and Toshihiko Komine, started to make use of them also to transmit data by using a technology, which was a hot topic that time, named power line communication (PLC) [26, 27]. This technique uses the electric network for both transmission of data and electric energy. The illustration of this process is shown in Figure 1.4, in which data are properly converted by the

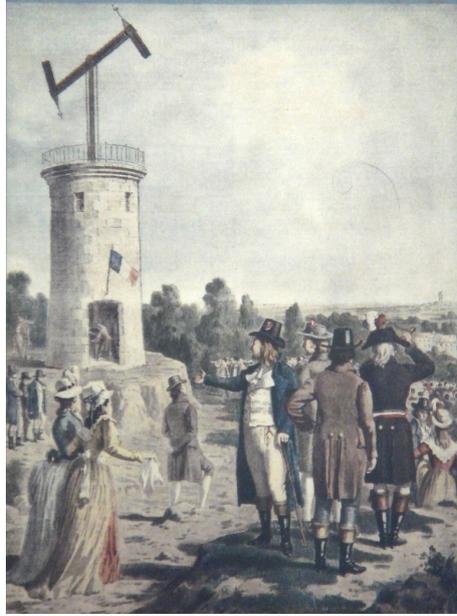


Figure 1.1: Illustration of the optical telegraph. Reprinted from [1].

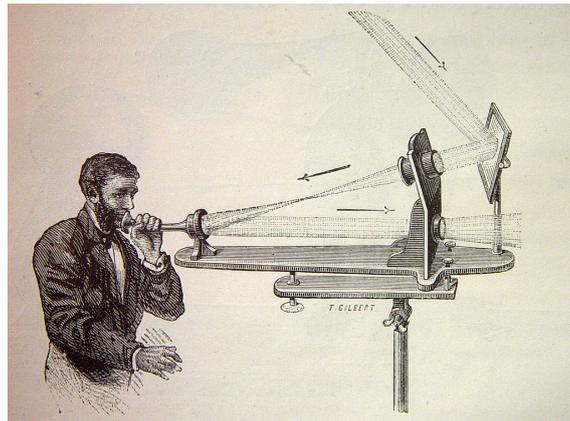


Figure 1.2: Illustration of the photophone transmitter. Reprinted from [2].

PLC modem and transmitted by LEDs.

Since the 2000s, lots of works in VLC popped up. In 2003, the research group led by Tanaka *et al.* presented a VLC system with a data rate up to 400 Mbps based on OOK modulation [28]. In 2005, the authors in [29] first proposed the use of OFDM over an indoor optical channel. In 2006, the first experimental results using OFDM were produced [30]. In 2009, a data rate of 230 Mbps was reached using discrete multitone (DMT), a single white LED luminary [31], and a PIN. The same team demonstrated a data rate of 513 Mbps by using a more powerful LED luminary and an avalanche diode [32]. In 2012, also using DMT and a single low-cost white LED, a data rate of 1 Gbps was reported [33]. Later, also in 2012, a data rate of 3.4 Gbps was achieved by employing wavelength division multiplexing (WDM) and a red-green-blue (RGB) LED [34]. In 2015, a data rate of 750 Mbps was achieved

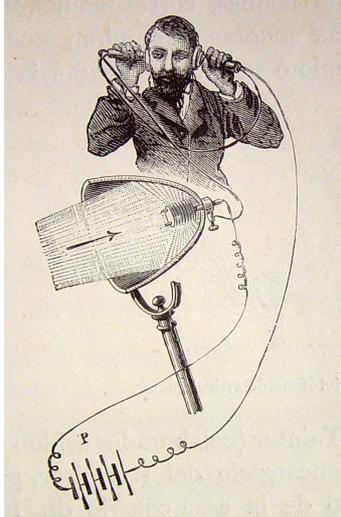


Figure 1.3: Illustration of the photophone receiver. Reprinted from [3].

using a RGB LED with analog equalization [35]. Also in 2015, a demonstration of 8 Gbps of data rate was reached by using a single luminary red-green-blue-yellow (RGBY) and WDM [36]. A 25 Gbps data rate was reported in 2016, but using laser diodes (LD), which are very expensive and emit a high focused energy beam [37]. In fact, until the present day of the writing of this work, the literature reports works achieving data rates in the order of Gbps employing LDs, which are costly and not employed for illumination purposes being, therefore, beyond the scope of this work.

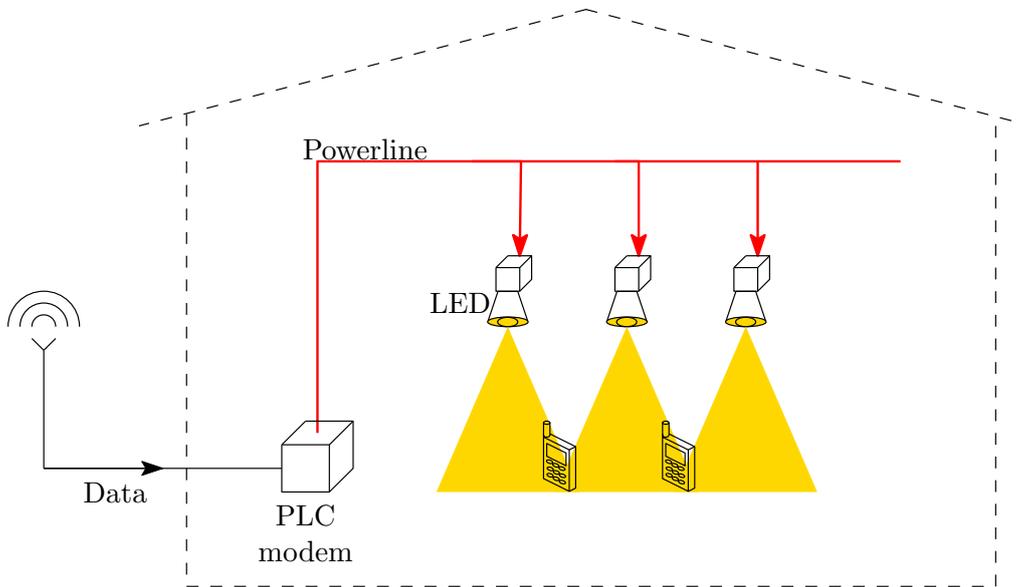


Figure 1.4: Illustration of PLC and VLC systems in an indoor environment.

### 1.3 VLC Main Challenges

VLC has been proved to feature several advantages. Nevertheless, there are still some challenges to overcome in this technology:

- Low light level: Considering an indoor environment which uses LED for both illumination and communication purposes, it may be undesirable a system which demands lights on all the time. Moreover, the shadowing effect, which may be caused by an obstacle between LED and photodiode, can cause problems in the communication process. Therefore, VLC systems should be able to maintain a minimum data rate when the light level is not perceptible to humans. In fact, that are studies conducted in this area, which reached satisfactory data rates considering a low level of ambient light [38], or employed infra-red (IR) signals when the light level is low [39].
- Flickering and average optical power: the signals in VLC should not present long sequences of “1s” or “0s”, since it may cause flicker effect perceptible for humans, which is not desirable in illumination purposes. Therefore, in order to avoid it, VLC systems should employ run length limited (RLL) sequences [40]. In addition, if one considers an indoor VLC deployment, the average optical power should maintain a constant or slow time-varying level of optical power. The aforementioned constraints may hamper the optimization process performed in precoding or beamforming schemes [41, 42].
- Nonlinear equalization: The current-voltage relation of LEDs is well modeled by an exponential function. Hence, this device may present a considerable level of nonlinearity, depending on the desired electrical power at output. Therefore, suitable nonlinear equalizers must be employed in order to have a reasonable bit error rate (BER) at the receiver [36, 43, 44]. Other techniques, such as pre- and post-distortion can also mitigate the nonlinear effects of a VLC system [45–47], but these techniques are designed based on the  $I$ - $V$  curve of a given LED at the transmitter, which can be a hard task if multiple LEDs are employed, as in multiple-input multiple-output (MIMO) systems.

### 1.4 Contributions of this Work

Considering the challenges described in Section 1.3, this work will focus on proposing nonlinear adaptive equalizers to mitigate the undesired nonlinear effects yielded by VLC systems. To do that, it will be employed the well-known Volterra series, which can also be used in a myriad of nonlinear systems, such as satellite and microwave

channel amplifiers [48, 49], nonlinear controllers [50–53], nonlinear echo cancellation [54, 55], just to mention a few. While kernel-based adaptive filtering has been attracting much interest in recent years [56–58], it still lacks a satisfactorily universal quantization rule to address its inherent unlimited memory requirements. In fact, most practical applications requiring online nonlinear solutions employ the celebrated Volterra series [59–61] due to its structural simplicity [62] and its capability of modeling a variety of nonlinear systems [63]. However, the main drawback of this approach is the computational burden given the large number of parameters to be updated, as compared to its linear-in-input counterparts. This fact may cramp the use of Volterra series in applications that require long adaptive filters or in the case of stand-alone embedded systems, which demand low energy consumption. Recent works reduce this computational complexity by employing a sparse scheme to the Volterra filter [64], by reducing the number of operations per parameter [65], or by using it in the frequency domain [66, 67], but none of them use a data-selective approach.

One of the contributions of this work reduces the computational costs related to nonlinear adaptive filters by proposing data-selective algorithms that employ Volterra series. By connecting the set-membership adaptive filtering (SMAF) [68, 69] framework to the Volterra series, it is possible to reduce drastically the computational cost during the training by selecting the input data that indeed brings innovation to the system, avoiding unnecessary updates during the training process, and keeping, or even enhancing, the original performance. The idea here is to employ, at each iteration, a constraint set and check whether the magnitude of the error between the desired (reference) signal and the output of the adaptive filter is under a given threshold, i.e., whether the parameter vector belongs to this constraint set. If this is the case, the parameter vector is not updated. The proposed techniques can be cast as Volterra-based versions of the set-membership proportionate normalized least-mean-square algorithm (SM-PNLMS) [70, 71], and of the bounding ellipsoidal adaptive constrained least-squares (BEACON) [72, 73].

Another contribution of this work aims to reduce even more the computational complexity due to Volterra series by the use of combination of filters [74, 75]. By inspecting its structure, one can analyze separately the kernels of such series, whose elements are related to the order of the products. For instance, the second-order kernel is assembled by the corresponding elements to the second-order products of Volterra series. Thus, one can employ different data-selective-based adaptive filters for each kernel and combine their parameter vectors to reduce the overall complexity of Volterra series. In addition, it is possible to reduce the amount of data employed during the training of these adaptive filters, which can use a considerable quantity given the large number of parameters to be updated when using Volterra series,

reducing the spectral efficiency. Therefore, the other proposal of this work is to use semi-blind data-selective algorithms [76], which enhance the spectral efficiency by replacing the reference signal with the decided version of the output of the adaptive filter, while reducing the computational burden associated with Volterra series.

The last contribution, is the development of a computational platform for VLC, which is employed on the assessment of the performance of the proposed data-selective techniques. An open-source platform for developing practical VLC solutions is detailed in [77]. However, to the best of our knowledge, no work focusing on developing an open-source simulation environment for this application had been presented.

## 1.5 Organization of the Text

Chapter 2 describes an overview of the operation mode of a VLC system, as well as its key components and their physical aspects. Regarding LEDs, the working principle of the *pn* junction is discussed, as well as their nonlinear current-voltage relation and frequency response. In addition, the main types of LEDs used in VLC white color-based systems are described with their optical characteristics. A description of the indoor free-space optical (FSO) channel is performed, detailing its frequency selectiveness and attenuation, also drawing a parallel with RF channels. To finish, the modes of operation of typical photodiodes employed in VLC systems are listed, including the current-voltage relation corresponding to each mode, and their spectral responses.

In Chapter 3, the Volterra set-membership adaptive filtering (SMAF) is proposed. Volterra filters are widely used to work with nonlinear systems. Nonetheless, this technique may yield a high computational complexity, if compared to its linear counterpart. Therefore, a good approach to address this issue is to connect a data-selective framework to Volterra series. Moreover, by using a generic nonlinear system, the performance of Volterra SMAF is assessed in both system identification and channel equalization scenarios, indicating that this technology indeed can be used to cope with generic nonlinear systems, while reducing the computational burden due to Volterra series. Therefore, this technique may be also employed in VLC systems as an efficient and reliable equalizer.

Chapter 4 presents a new proposed algorithm, which explores the structure of the Volterra series to reduce even more the computational complexity associated with it. Besides, a semi-blind technique is employed in the Volterra SMAF, where the goal is to decrease the amount of data used to train the adaptive filters, thereby, enhancing the spectral efficiency. The performance of these techniques are, in general, similar or better than the one of Chapter 3. Hence, these techniques are also a good bet to

work as equalizers for VLC.

Chapter 5 describes a computational platform for VLC, whose main purpose is to serve as tool for the performance assessment of the proposed adaptive techniques presented in the previous chapters. In addition, based on their physical description, as performed in Chapter 2, the models of the main devices of such system are presented.

By employing the computational platform developed in Chapter 5, the performance of the proposed techniques is evaluated in Chapter 6, which is assessed in terms of mean square error (MSE) and bit error rate (BER). Moreover, discussions regarding these results are performed.

Finally, in Chapter 7, the conclusions obtained from the results are drawn, as well as discussions regarding all the proposed techniques, and future works that may continue this project.

The notation used in this work is as follows: vectors are denoted by bold letters while matrices are represented by capital bold letters.  $(\cdot)^T$ ,  $(\cdot)^H$ , and  $(\cdot)^*$  denote the transpose, Hermitian, and conjugate operations, respectively.  $\mathbb{N}$ ,  $\mathbb{R}$ , and  $\mathbb{C}$  denote the natural, real, and complex sets.  $\|\cdot\|_1$ ,  $\|\cdot\|_2$ , and  $\|\cdot\|_\infty$  stand for the  $l_1$ -norm,  $l_2$ -norm, and infinite-norm. Considering  $b \in \mathbb{N}$ , the sign function is defined as  $\text{sign}_b[a] = \frac{a}{\|a\|_b}$ ,  $\forall a \in \mathbb{C} \setminus \{0\}$  and  $\text{sign}_b[0] = 0$ .

## 1.6 Publications

This section lists the published works as well as papers that are in preparation and which resulted from this dissertation:

- da Silva, F. B., Martins, W. A.: A computational platform for visible light communications. In: “*XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2017)*”, pp. 891–895. São Pedro, Brazil (2017).
- da Silva, F. B., Martins, W. A.: Data-Selective Volterra Adaptive Filters, “*Circuits, Systems, and Signal Processing*”, Jan 2018.
- Journal paper in preparation containing the contributions of Chapter 4.
- Journal paper in preparation containing the contributions of Chapter 6, along with the descriptions of Chapters 2 and 5.

# Chapter 2

## VLC Main Components

The key components of any digital communication system are: (i) transmitter, which processes the data so as to pass through the communication medium; (ii) the medium (or channel, in the engineering context) itself, which usually imposes undesired effects on the signals after transmission; and (iii) the receiver, which converts the received signals into data that are supposed to be good estimates of the transmitted data. In a visible light communication (VLC) system, the aforementioned elements are the light-emitting diode (LED), the optical channel, and the photosensor, commonly a photodiode (PD). In this chapter, an overview of a VLC system is presented. After that, in order to understand with more details the working principles underlying VLC systems, this chapter describes the main characteristics of each of these components.

### 2.1 VLC System Overview

In a nutshell, the operation mode of a VLC system is quite simple. The data to be transmitted is mapped by the driving circuit into an electrical signal  $V_{\text{in}}(t)$ , which is converted to light using an LED. This light signal  $l(t)$  propagates through the air (free space) until it reaches the receiver where it is transformed over again into the electrical signal using a receiver sensor, and then, amplified via a transimpedance amplifier (TIA), generating the voltage signal  $r(t)$ . This process, illustrated in Figure 2.1, resembles what occurs in radiofrequency (RF) communications, except for the electrical conversion to light at the transmitter and the inverse process at the receiver. Nevertheless, in the RF framework the amplitude of the received electrical signal is usually a linear function<sup>1</sup> of the electrical field, instead of nonlinear as in the VLC case [17]. VLC schemes employ intensity modulation (IM), where the data to be transmitted is modulated varying the luminous intensity of the emitted

---

<sup>1</sup>There are some exceptions due to nonlinearities induced by power amplifiers employed in the transmission process.

light [17]. At the receiver, the fluctuations in the light luminosity are converted into a proportional current level by the photodiode. This process is the so-called direct detection (DD) [17]. Generically speaking, this process of transmitting and receiving data through a VLC system is described as follows: first, the electrical-to-optical conversion is performed by the LED, here represented as an operator  $L\{\cdot\}$ :

$$l(t) = L\{V_{\text{in}}(t)\}. \quad (2.1)$$

After that, the optical channel, represented mathematically as  $H\{\cdot\}$ , transforms the signal intensity of  $l(t)$ , resulting in the signal  $\tilde{l}(t)$ :

$$\tilde{l}(t) = H\{l(t)\}. \quad (2.2)$$

Then, the output of the transimpedance amplifier can be described as:

$$r(t) = \beta \times R\{\tilde{l}(t)\}, \quad (2.3)$$

where  $R\{\cdot\}$  denotes the optical-to-electrical conversion performed by the photodiode, and  $\beta \in \mathbb{R}_+$  represents the amplification gain induced by the TIA.

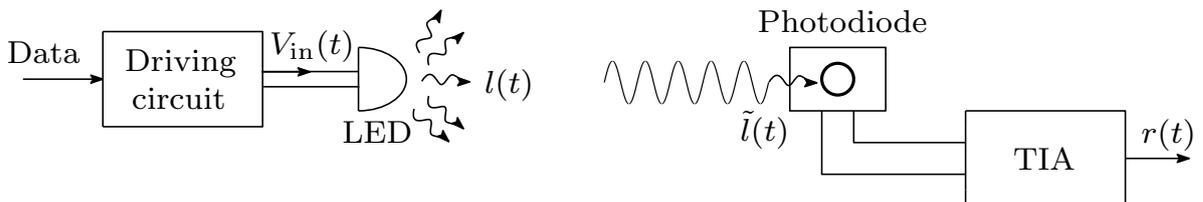


Figure 2.1: Overview of a VLC system.

## 2.2 Light-Emitting Diode

The LED is a device which emits light when it is properly polarized. In order to better understand its physical properties, some basic concepts of semiconductors shall be described. This device comprises a  $pn$  junction, which is compounded by semiconductors of  $p$  and  $n$  types connected to the LED's anode and cathode, respectively. In practice, this junction is assembled by a single crystal with different doping levels [78].

Considering an intrinsic semiconductor, the atoms in a crystal are bonded via covalent bonds, assembled by the four valence electrons associated with each atom. In low temperatures, only few electrons are available to conduct electricity, but at common indoor temperatures ranging from 20° C to 25° C, some bonds break, thereby freeing up electrons. As these electrons escape from their atoms, they leave

gaps conceived as positive charges. Other free electrons are attracted by these positive charges. This process, named recombination, of releasing electrons and generating gaps (or holes) may repeat itself through the crystal, allowing for a current flow. Nonetheless, this flow is usually very low due to the number of free electrons in an intrinsic crystal. In order to address this issue, the crystal is doped with chemical elements containing three or five valence electrons. Thus, when these elements are bonded, an electron is released (*n*-type crystal), or a hole rises (*p*-type crystal), but in a much greater rate. For further details, the references [78, 79] should be consulted.

In fact, the LED is a semiconductor that emits photons when the recombination process in its *pn* junction occurs, i.e, when current flows from the anode to the cathode.<sup>2</sup> Therefore, this device may be employed as transmitter in a VLC system, and its main purpose is to convert electrical signals into optical signals.

Regarding to the current-voltage *I-V* curve of an LED, this relation is highly nonlinear, depending on the amplitude of the input signal, which is why the LED may cause severe distortion over the transmitted signal. Drawing a parallel with RF systems, in most applications the nonlinearities of the system can be disregarded, and the whole system can be considered as linear. Nonetheless, satellite and microwave systems may experience a considerable level of nonlinearities due to the power amplifier at the transmitter's output [48, 49]. Figure 2.2 illustrates a typical *I-V* curve of an LED, also showing the regions in which the LED works: forward, reverse, and breakdown regions. In the forward region, this is the region in which the LED emits light, i.e, when current flows from the anode to its cathode. In the reverse region, only the saturation current (or dark-current) flows in the opposite direction of the forward case. The breakdown region is reached when a negative voltage  $V_{br}$  is applied in LED's terminals, harming the *pn* junction, which can impair the light emission, even though there are other types of diodes that are designed to work in this region, such as Zener.

With respect to the propagation of the light emitted by an LED, it is much more focused than in the radiofrequency case. Therefore, the interference among VLC transmitters are rare or absent in this technique, which allows for the reuse of the spectrum. Nonetheless, it requires a certain level of alignment between transmitter and receiver, whose communication may be affected if obstacles are placed in-between. In a RF scenario, omnidirectional antennas are usually employed, which may yield a considerable level of interference with other signals, but makes it easy the deployment of receiving antennas.

The response of LEDs for each wavelength of the transmitted signal should be

---

<sup>2</sup>The flow direction can appear from the cathode to the anode, depending on the convention adopted.

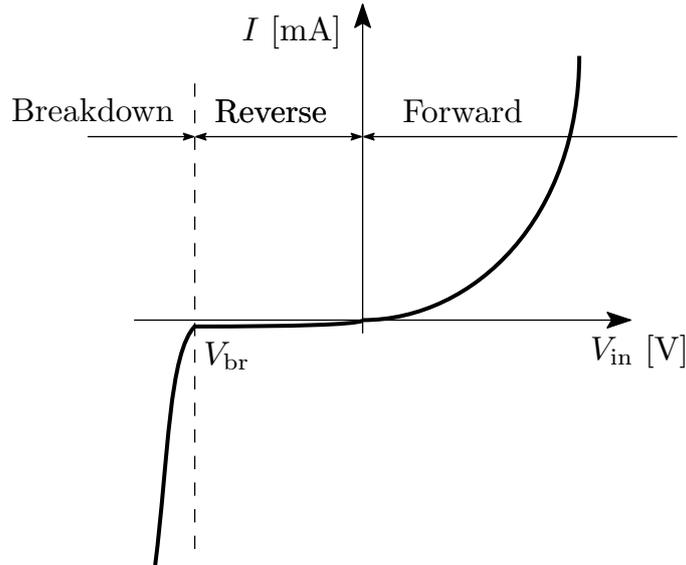


Figure 2.2: Typical LED  $I$ - $V$  curve.

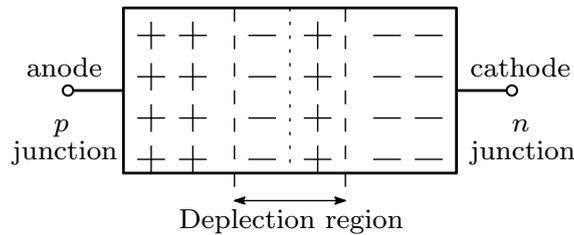


Figure 2.3: Illustration of the depletion region of a  $pn$  junction.

carefully analyzed. In an indoor VLC system, where the white color is widely employed, two types of LEDs are employed to implement white color-based communications: phosphor-converted LED (pc-LED) and the three-colored red-green-blue, or RGB LED. Regarding the former, a blue LED is filled with a phosphor layer. Basically, the working principle is to leak some part of the blue light to the phosphor so as to generate red and green colors, combining these colors with the blue one to form white light [80]. Considering the RGB LED, it directly mixes the red, blue, and green lights to assemble the white color. The shape of the spectral response of the aforementioned LEDs are shown in Figure 2.4.

Another important characteristic of LEDs is their frequency response. Supposing that  $V_{in}(t)$  is a high frequency signal, it induces a high frequency current flow over the LED's  $pn$  junction. Nonetheless, during the recombination process, some free electrons disappear from the  $n$  junction, thus some of the positive charges will not be recombined with these free electrons. Then, these positive charges will attract the negative charges of the  $p$  junction, forming the depletion region, illustrated in Figure 2.3. This region acts as a capacitor, which imposes a memory effect on the system and may cease or strongly attenuate the current flow over the  $pn$  junction. Thus, the LED cannot always light up and turn off in the same frequency of the

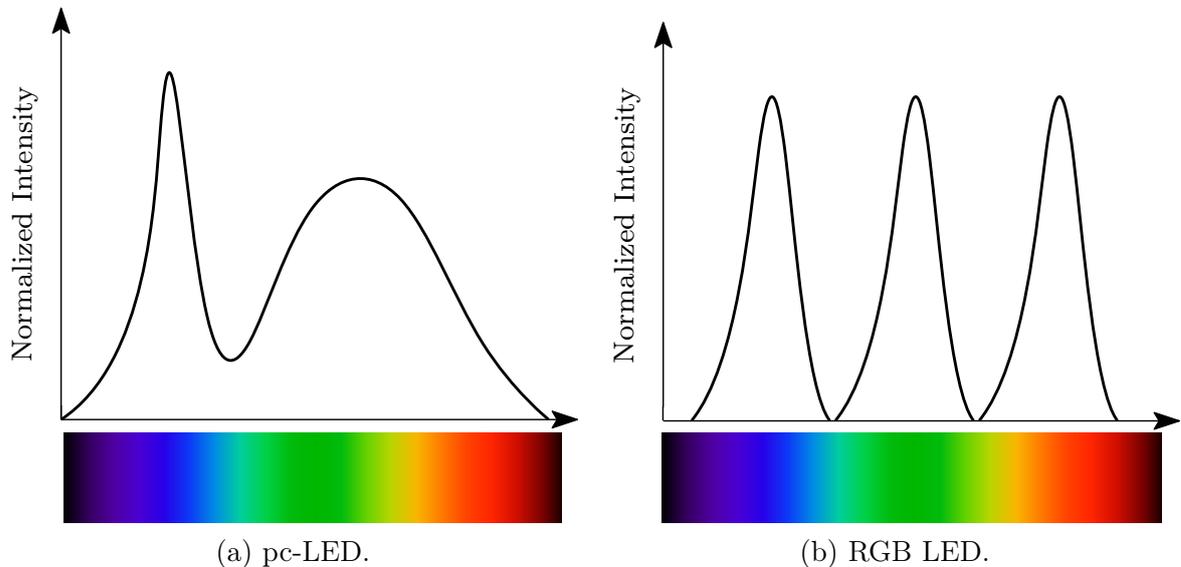


Figure 2.4: Sketches of the spectral response (in terms of wavelength) of LEDs employed in white color-based communications.

input signal  $V_{in}(t)$ . Moreover, the frequency response of LEDs may be affected by the way they generate the colors. For instance, when pc-LEDs generate the white color, part of the blue light is leaked to the phosphor layer. Due to its slow response, a memory effect pops up, decreasing the frequency response of the LED. In order to achieve wider bandwidths, blue filter is employed at the photodiode to filter out the yellow component (570–590 nm), i.e., to attenuate the photons that passed through the phosphor layer [81], leading to modulation bandwidth up to 30 MHz [32, 33]. However, it comes at the price of decreasing the signal-to-noise ratio (SNR), as the major portion of the received signal is also filtered out [82]. On the other hand, RGB LEDs can achieve modulation bandwidths wider than 40 MHz [7].

## 2.3 Free-Space Optical Channel

The free-space optical (FSO) channel is the medium through which the visible light emitted by the LED propagates. In most practical indoor environments, the visible light is much more attenuated by obstacles, such as walls and furniture, than RF signals, implying that the FSO channel presents low levels of frequency selectivity, resulting in a relatively flat frequency response [17]. In fact, some studies show that the FSO channel resembles an infrared (IR) channel [17], and most works in the literature consider IR channel models when dealing with VLC systems, given their high similarities in terms of propagation properties and sources of noise. Thus, considering a flat frequency response, the most relevant characteristic of the optical channel is its DC gain  $H(0)$ . Depending on the geometry of the environment in which the VLC system is deployed, i.e., walls and obstacles between LED and photodiode,

$H(0)$  may represent a large attenuation over the LED's emitted light. Besides, in vehicle-to-vehicle communication, for instance, the turbulence may also impair the communication in a VLC system. It is worth mentioning that, in the optical communication area, every element which introduces some attenuation or distortion between LED and photodiode is considered to be part of an effective channel; the effects imposed by light concentrators or optical filters in photodiodes are a case in point. Comparing with RF-based communication systems, which may present high frequency selectivity and strong attenuation, the FSO channel may also pose a considerable attenuation, but with low frequency selectivity due to the severe light absorption of the obstacles and the relatively small indoor dimensions.

Regarding ambient noise sources, if one considers an indoor VLC deployment, there are many sources of light that may interfere with the communication between LED and photodiode. Sunlight, incandescent and fluorescent lamps, or even other LEDs whose only purpose is illumination can work as interference sources [83]. Regarding other noises, one can mention the shot and thermal noise. Considering the former, when photons are emitted by the LED, they do not strike the photodiode surface continuously in time, instead, they arrive erratically, causing fluctuations over the resulting current at the photodiode. As for the thermal noise, also named Johnson-Nyquist or  $1/f$  noise, it is related to the thermal agitation of the charge carriers inside an electrical conductor, which produces a current flow over the photodiode and TIA and it is highly dependent on the ambient temperature. Most of the aforementioned noise sources are also faced by an RF system. For instance, the sun emits several electromagnetic signals that could interfere with RF signals. In addition, the shot noise also occurs in other devices which possess a  $pn$  junction, such as transistors, and the thermal noise is inherent in diverse electronic components, e.g., resistors and capacitors.

## 2.4 Photodiode

The photodiode (PD) is a device that performs the inverse function of the LED: incident light is converted into current. As in the LED's case, PD is compounded by a  $pn$  junction connected between anode and cathode, but, unlike LEDs, it can operate in two modes:

- Photoconductive mode: PD is reversely polarized by an external voltage source, as illustrated in Figure 2.5(a). In this case, the capacitance of the  $pn$  junction decreases due to a large depletion region, which is caused by the reverse-bias [84]. Thus, the photodiode becomes more capable of working with

broadband signals at a price of larger dark-current.<sup>3</sup>

- Photovoltaic mode: forward-bias takes place in this case, as illustrated in Figure 2.5(b). This mode is based on the photovoltaic effect, which is a physical property of creation of a current flow in a material upon exposure to light. Basically, supposing the PD's terminals are connected, electrons absorb the energy of incident photons, starting the recombination process, thereby generating a current flow. This is the working principle of solar cells.

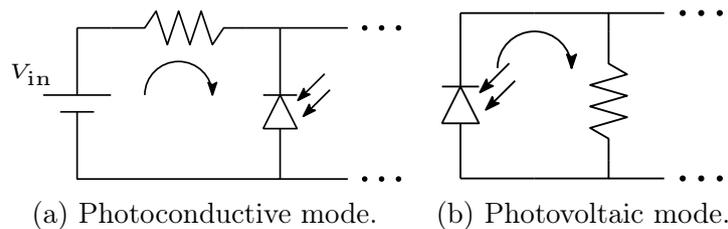


Figure 2.5: Photodiodes' modes of operation.

The current-voltage  $I$ - $V$  relation of photodiodes is very similar to the LED's. Figure 2.6 illustrates this characteristic curve for both photoconductive and photovoltaic modes and considering different incident powers, denoted as  $P_0$ ,  $P_1$ , and  $P_2$ , with  $P_0 < P_1 < P_2$ . Note that as the incident power grows, the resulting current is almost constant in the reverse mode, thus, the relation incident power vs. current is practically linear. This, fact along with the capability of working in a high frequency scenario are the main reasons why most VLC systems adopt this mode. It is worth mentioning that in both modes the output current is in the order of micro amperes. Hence, for most applications a transimpedance amplifier (TIA), which converts a level of current into a level of voltage, is needed.

Regarding the types of photodiode, two of them are widely employed in VLC systems: a diode composed of an intrinsic region between the  $pn$  junction (PIN) and avalanche photodiodes (APD). Essentially, the main difference between them is that the avalanche photodiode operates at high reverse-bias, resulting in an internal electrical gain due to avalanche breakdown [17], eventually making it more suitable for low levels of ambient light [85]. Nonetheless, this gain increases the variance of shot noise by a factor greater than the signal gain, reducing the SNR [17]. Therefore, the PIN photodiode is the most employed in VLC system, besides being cheaper than APD.

Regarding the frequency response of photodiodes, in general, it has wider bandwidth than the LED's [86]. In fact, the overall frequency response at the receiver is determined by both photodiode and transimpedance amplifier, even though the

<sup>3</sup>Dark-current is the small current that flows in the photodiode in the absence of light.

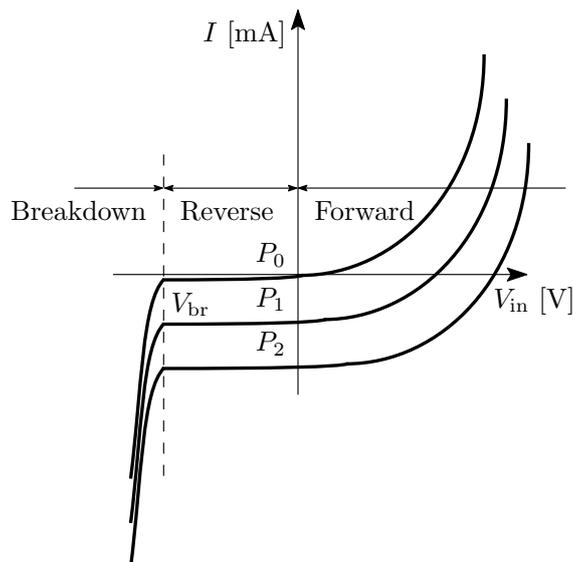


Figure 2.6: Characteristic  $I$ - $V$  curve in reverse-bias mode for different values of incident power.

TIA's has wider bandwidth than the LED's. Due to these facts, the main source of selectivity can be attributed to LEDs, at least considering low data rate communications.

Typical photodiodes used in VLC have spectral peaks around 800 nm, as shown in Figure 2.7; thereby the sensitivity is higher to red light than green or blue [28]. In fact, the authors in [87] show that the performance of a VLC system that employs RGB LEDs is color-dependent, i.e., it shows different performance for each one of the three colors, yielding the highest data rates for the red color. It is worth pointing out that ambient light optical power induces shot noise at the photodiode. Besides, when TIA is employed, thermal noise is produced. In fact, if one considers the ambient noises discussed in Section 2.3, and in this subsection, the overall resulting noise at the photodiode can be considered white and Gaussian [88].

In some applications, other components, such as optical filters and concentrators may be coupled to the PD. The function of the optical filter is to attenuate optical signals whose wavelength differs from the filter. For instance, if one is employing an RGB LED, three photodiodes may be employed at the receiver, each one of them selecting the red, green, and blue colors. The objective of the concentrator filter is to filter in the space domain, increasing the necessity of an alignment between LED and PD, but it reduces the effect of punctual noise sources, as other LEDs or lamps.

As described during this chapter, the key components of a VLC system may yield a considerable level of nonlinearity, calling for a nonlinear equalizer. Besides, as discussed in Section 2.2, the LED exhibits a memory effect due to the depletion region of its  $pn$  junction, demanding for equalizers with memory. In order to address these two requirements, this work proposes some nonlinear adaptive techniques in

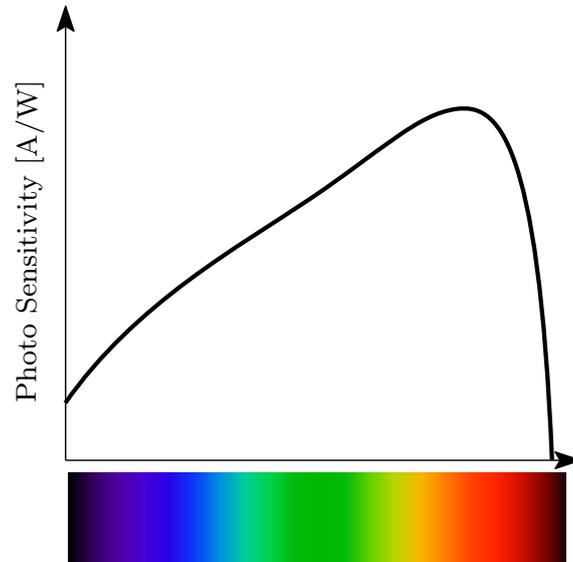


Figure 2.7: Typical shape of spectral response of photodiodes.

Chapters 3 and 4. In addition, the main components of a VLC system presented in this chapter serve as background for modeling each element of a computational platform for a VLC system, which will be presented in Chapter 5. Thus, the performance of the proposed methods can be assessed properly within a VLC scenario in Chapter 6.

# Chapter 3

## Data-Selective Volterra Adaptive Filters

This chapter proposes the use of the data-selective framework within the context of Volterra adaptive filters. It starts by describing the concept of adaptive filtering and the well-known data-selective adaptive algorithms, such as the proportionate version of the normalized least-mean-squares (NLMS) and the recursive least-squares (RLS). Adaptive filters adapt their parameter vector according to a given function of the output error, and are suitable when working with time-variant systems. When these systems are nonlinear, Volterra series is one of the most popular techniques to cope with the nonlinearities. However, the main drawback of Volterra-based filters is the computational burden given the large number of parameters to be updated, as compared to its linear-in-input counterparts. In order to mitigate this effect, the set-membership adaptive filtering framework (SMAF) is connected to the Volterra series, reducing drastically the computational cost by selecting the input data that indeed brings innovation to the system, avoiding unnecessary updates during the training process, and keeping, or even enhancing, the original performance.

### 3.1 Adaptive Filtering

An adaptive filter is a system that can be parameterized by a parameter vector  $\mathbf{w} \in \mathbb{C}^{L \times 1}$  of length  $L \in \mathbb{N}$ , which can change depending on the value of a given function of the output error  $e[k] \triangleq d[k] - y[k]$ , with  $y[k] = \mathbf{w}^H[k] \mathbf{x}[k]$ , where  $d[k] \in \mathbb{C}$  represents the desired (or reference) signal, and  $\mathbf{x}[k] \in \mathbb{C}^{L \times 1}$  is a vector containing the input signal at the  $k^{\text{th}}$  iteration. The general configuration of an adaptive filter is illustrated in Figure 3.1. Adaptive filters are usually employed when the design specifications of the filter are unknown or time-varying; hence, it updates its parameter vector employing a reference signal, which depends on the application. In

a system identification scenario, the objective is to model a unknown system using an adaptive filter, whose scheme is illustrated in Figure 3.2.

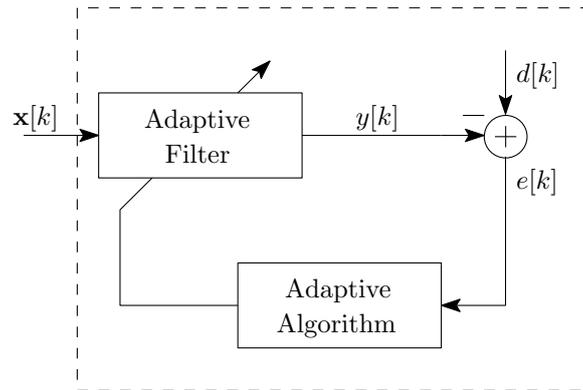


Figure 3.1: General configuration of an adaptive filter.

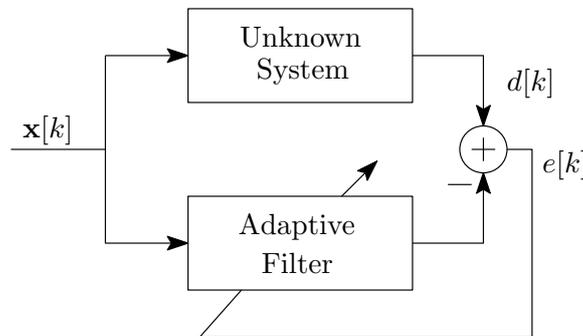


Figure 3.2: System identification scheme.

Regarding the channel equalization application, the goal is to mitigate the distortions that the channel yields by employing an adaptive filter, where the input signal, delayed by  $D \in \mathbb{N}$  samples, plays the role as reference signal, as shown in Figure 3.3. It is worth mentioning that other applications, such as signal prediction, signal enhancement, and echo cancellation are also widely explored in the context of adaptive filtering [68].

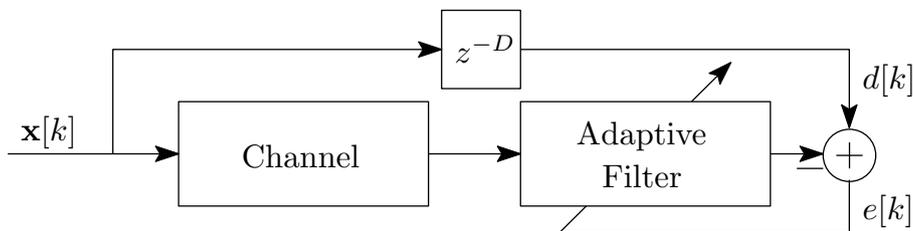


Figure 3.3: Channel equalization scheme.

## 3.2 Linear-in-input SMAF

The SMAF technique aims to provide a parameter vector  $\mathbf{w}$  such that the magnitude of the output error is upper bounded by a predefined constant  $\bar{\gamma} \in \mathbb{R}_+$ , for all available pairs  $(\mathbf{x}, d)$ . The set of all vectors  $\mathbf{w}$  satisfying  $|e| \leq \bar{\gamma}$  is called *feasibility set* [68, 69], denoted as

$$\Theta = \bigcap_{\forall(\mathbf{x}, d)} \{\mathbf{w} : |d - \mathbf{w}^H \mathbf{x}| \leq \bar{\gamma}\}. \quad (3.1)$$

In real time applications, a pair  $(\mathbf{x}[k], d[k])$  is available at each discrete-time index  $k \in \mathbb{N}$ . Thus, computing  $\Theta$  would require an infinitely large memory. Nevertheless, the set  $\Theta$  can be assembled in an iterative fashion using only the available data pairs up to the  $k^{\text{th}}$  iteration. Considering the *constraint set*

$$\mathcal{H}[k] = \{\mathbf{w} : |d[k] - \mathbf{w}^H \mathbf{x}[k]| \leq \bar{\gamma}\} \subset \mathbb{C}^L, \quad (3.2)$$

which includes all possible vectors  $\mathbf{w}$  at the  $k^{\text{th}}$  iteration whose corresponding error magnitudes are bounded by  $\bar{\gamma}$ , and a set

$$\Psi[k] = \bigcap_{i=0}^k \mathcal{H}[i], \quad (3.3)$$

one may recursively estimate  $\Theta$ , since  $\lim_{k \rightarrow \infty} \Psi[k] = \Theta$ .

### 3.2.1 SM-PNLMS

Some of the most common adaptive algorithms are the least-mean-squares (LMS) [68, 89] and its normalized version (NLMS) [68] thanks to their computational simplicity and robustness. These algorithms update their parameter vectors using the direction indicated by the input vector. However, if the underlying system can be approximated through a sparse model, it is more efficient to update using a different direction that can be defined based on the magnitude of the coefficients of the current parameter vector. The *set-membership proportionate-NLMS (SM-PNLMS)* [70, 71] implements this proportional update strategy, besides employing a data-selective scheme to reduce the computational complexity. This algorithm updates according to the minimal perturbation principle, described as [71]

$$\begin{aligned} \mathbf{w}[k+1] &= \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \\ &\text{subject to : } d[k] - \mathbf{w}^H \mathbf{x}[k] = \bar{\gamma} \text{sign}\{e[k]\}, \end{aligned} \quad (3.4)$$

where  $\mathbf{G}[k] \in \mathbb{R}^{L \times L}$  is a diagonal weighting matrix. Denoting  $\kappa \in [0, 1]$  as a sparse information factor, one can express the update equation of the  $l^{\text{th}}$  diagonal entry of  $\mathbf{G}[k]$  as [71]

$$\mathbf{G}_{ll}[k] = \frac{1 - \kappa\mu[k]}{L} + \frac{\kappa\mu[k]|\mathbf{w}_l[k]|}{\|\mathbf{w}[k]\|_1}, \quad (3.5)$$

where  $\mathbf{w}_l[k]$  denotes the  $l^{\text{th}}$  entry of  $\mathbf{w}[k]$  and

$$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{|e[k]|}, & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.6)$$

After solving the optimization problem described in (3.4), it is possible to express a feasible update for the SM-PNLMS as

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu[k] \frac{\mathbf{G}[k]\mathbf{x}[k]}{\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]} e^*[k]. \quad (3.7)$$

Then, the vector  $\mathbf{w}[k]$  updates in the direction of the input vector, but proportionately with the magnitude of each entry of  $\mathbf{x}[k]$ . In addition, the energy of the input signal is normalized by the factor  $\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]$ . The derivation of the previous update equation can be found at Appendix A.1, whereas the pseudo-code of this technique is described in Appendix B.1. The generalization of this derivation for the *affine projection* algorithm [68, 90] can be found in [71].

### 3.2.2 Modified BEACON

Another popular adaptive algorithm is the recursive least-squares (RLS) [68, 91], which features fast convergence and good tracking performance when working in time-varying environments, as compared to LMS-based algorithms. These good properties are achieved at the expense of computational complexity. In order to overcome this problem, the authors in [72] proposed a data-selective version of the RLS, namely the *bounding ellipsoidal adaptive constrained least-squares (BEACON)* algorithm, which belongs to the class of optimal bounding ellipsoid (OBE) algorithms [92, 93]. This technique also estimates the feasibility set in (3.2) using an iterative procedure, though via a different approach. Basically, the main idea is to enclose the feasibility set at each iteration using an ellipsoid  $\mathcal{E}[k]$  such that

$$\mathcal{E}[k] \supset (\mathcal{E}[k-1] \cap \mathcal{H}[k]) \supset \Psi[k], \quad (3.8)$$

where  $\mathcal{E}[k]$  is an ellipsoid centered at  $\mathbf{w}[k]$ , parameterized by a Hermitian matrix  $\mathbf{S}[k]$ . To perform such operation, this algorithm updates according to the following

cost function:

$$\begin{aligned} \mathbf{w}[k] = \arg \min_{\mathbf{w}} \cdot & \left( \sum_{i=1}^k \lambda^{k-i}[k] |d[i] - \mathbf{w}^H \mathbf{x}[i]|^2 \right) \\ \text{subject to : } & |d[k] - \mathbf{w}^H \mathbf{x}[k]|^2 = \bar{\gamma}^2, \end{aligned} \quad (3.9)$$

where  $\lambda[k]$  plays the role of forgetting factor and it is optimized within the optimization process [94]. The *a priori* error is here defined as  $e[k] = d[k] - \mathbf{w}^H[k-1]\mathbf{x}[k]$ . Thus, the resulting update equations are described as

$$\lambda[k] = \begin{cases} \frac{1}{\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \left( \frac{|e[k]|}{\bar{\gamma}} - 1 \right), & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise,} \end{cases} \quad (3.10)$$

with the update equation for  $\mathbf{S}[k]$  being expressed as

$$\mathbf{S}[k] = \mathbf{S}[k-1] + \lambda[k]\mathbf{x}[k]\mathbf{x}^H[k]. \quad (3.11)$$

By making use of the matrix inversion lemma [95], one can rewrite (3.11) and express the BEACON update equations as

$$\mathbf{S}^{-1}[k] = \mathbf{S}^{-1}[k-1] - \frac{\lambda[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]}, \quad (3.12)$$

$$\mathbf{w}[k] = \mathbf{w}[k-1] + \lambda[k]\mathbf{S}^{-1}[k]\mathbf{x}[k]e^*[k]. \quad (3.13)$$

The derivation of these update equations can be found at Appendix A.2.

The term  $\lambda[k]\mathbf{x}[k]\mathbf{x}^H[k]$  in (3.11) increases the norm of  $\mathbf{S}[k-1]$ . Hence,  $\lambda[k]$  tends to a large number as  $\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]$  tends to zero, leading to numerical instabilities. In order to cope with this issue, the authors in [73] proposed modifications in the BEACON recursion. By multiplying both sides of (3.11) by  $\lambda^{-1}[k]$ , results in

$$\lambda^{-1}[k]\mathbf{S}[k] = \lambda^{-1}[k]\mathbf{S}[k-1] + \mathbf{x}[k]\mathbf{x}^H[k]. \quad (3.14)$$

Defining as  $\tilde{\mathbf{S}}[k] = \lambda^{-1}[k]\mathbf{S}[k]$ , (3.14) can be rewritten as

$$\tilde{\mathbf{S}}[k] = \tilde{\mathbf{S}}[k-1] + \mathbf{x}[k]\mathbf{x}^H[k]. \quad (3.15)$$

By using the matrix inversion lemma, one can rewrite (3.15) as

$$\tilde{\mathbf{S}}^{-1}[k] = \lambda[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \frac{\tilde{\mathbf{S}}^{-1}[k] \mathbf{x}[k] \mathbf{x}^H[k] \tilde{\mathbf{S}}^{-1}[k-1]}{\lambda^{-1}[k] + \mathbf{x}^H[k] \tilde{\mathbf{S}}^{-1}[k-1] \mathbf{x}[k]} \right), \quad (3.16)$$

$$\mathbf{w}[k] = \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}[k] e^*[k], \quad (3.17)$$

where  $\lambda[k]$  is redefined as

$$\lambda[k] = \begin{cases} \mathbf{x}^H[k] \tilde{\mathbf{S}}^{-1}[k-1] \mathbf{x}[k] \frac{1}{\left(\frac{|e[k]|}{\bar{\gamma}} - 1\right)}, & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.18)$$

Thus, the parameter vector  $\mathbf{w}[k]$  updates in the directions defined by  $\tilde{\mathbf{S}}^{-1}[k]$ . Moreover, when the norm of  $\mathbf{S}^{-1}[k-1]$  increases,  $\lambda[k]$  decreases, reducing the norm of  $\mathbf{S}^{-1}[k]$ . The modifications imposed on the BEACON generate a feedback mechanism, enhancing the numerical stability of the algorithm. This technique will be named *Modified BEACON* (M-BEACON) in this text. In addition, the derivation of the preceding update equations can be found at Appendix A.3. A pseudo-code of the M-BEACON technique is described further in Appendix B.2.

### 3.3 Volterra-Based SMAF

Several practical systems perform nonlinear operations over the inputs to yield the output (e.g., satellite amplifiers [96], nonlinear controllers [50–53], and visible light communication systems [97, 98]). By truncating a series expansion of the output as function of the input signals, one can represent the nonlinear system behavior as a function of weighted combinations of products of the input samples [63, 68]. The Volterra scheme performs exactly this task: it applies a nonlinear transformation to the adaptive filter input in such a way that its output depends on combinations of input products. In the adaptive Volterra filtering context, the input follows a Volterra series truncated at the  $N^{\text{th}}$  order [68], generating the expanded vector  $\mathbf{x}_N[k]$ , with  $N \in \mathbb{N}$ . Considering a memory size of  $M \in \mathbb{N}$ , the  $N^{\text{th}}$ -order Volterra input vector can be recursively assembled as

$$\mathbf{x}_N[k] = [\mathbf{x}_{N-1}^T[k] \ \bar{\mathbf{x}}_N^T[k]]^T, \quad (3.19)$$

for  $N \geq 2$ , where the first-order Volterra series input vector  $\mathbf{x}_1[k]$  is described as  $\mathbf{x}_1[k] = [x[k] \ x[k-1] \ \cdots \ x[k-M]]^T$ , and each one of the  $\binom{N+M}{N}$  entries of  $\bar{\mathbf{x}}_N[k]$  is as follows:

$$x[k-m_1] \times x[k-m_2] \times \cdots \times x[k-m_N] \in \mathbb{C}, \quad (3.20)$$

for  $0 \leq m_1 \leq m_2 \leq \dots \leq m_N \leq M$ . Note that  $\mathbf{x}_N[k]$  does not contain redundant entries, so that the resulting length  $L$  of  $\mathbf{x}_N[k]$  can be expressed as [64]

$$L = \frac{(M + 1 + N)!}{(M + 1)!N!} - 1. \quad (3.21)$$

Hence, the ratio  $R_L \in \mathbb{R}_+$  between the  $N^{\text{th}}$ -order Volterra and linear-in-input filter lengths is

$$R_L = \frac{(M + 1 + N)! - (M + 1)!N!}{[(M + 2)! - (M + 1)!]N!}. \quad (3.22)$$

Unlike linear-in-input filters, the length of Volterra filters increases nonlinearly with the memory size or the order of the series expansion, which may induce a high computational burden for large memory/order. For instance, for  $N = 2$  and  $M = 2$ , the second-order Volterra input vector length is three times larger than the length of its linear-in-input counterparts. Therefore, the proposal of this work relies on employing the data-selective framework in Volterra filters to mitigate the aforementioned computational burden. Using (3.2), one can employ the  $N^{\text{th}}$ -order Volterra input vector to form the new Volterra constraint set, which is described by

$$\mathcal{H}_{\text{Volterra}}[k] = \{\mathbf{w} : |d[k] - \mathbf{w}^H \mathbf{x}_N[k]| \leq \bar{\gamma}\} \subset \mathbb{C}^L. \quad (3.23)$$

Since the adaptive filter updates only if  $\mathbf{w}[k] \notin \mathcal{H}_{\text{Volterra}}[k]$ , the computational complexity of Volterra adaptive filters can eventually be drastically reduced, as corroborated by the results in Section 3.4.

The proposed *Volterra SM-PNLMS* (VSM-PNLMS) and *Volterra M-BEACON* (VM-BEACON) filters are obtained by using  $\mathbf{x}_N[k]$  as input vector in the update equations described in Subsections 3.2.1 and 3.2.2. At this point, it is worth making a comparison between Volterra and kernel-based filtering in order to put the contributions of this chapter into a proper perspective. In the past decade or so, many kernel-based adaptive filters have been proposed. All of these new nonlinear systems rely on using well-known adaptive algorithms originally devised to work with linear-in-input systems. The key novelty of kernel-based filtering is the nonlinear mapping of the input vector into an infinite dimensional reproducing kernel Hilbert space (RKHS) [56]. Similarly, Volterra-based filtering also performs a nonlinear mapping of the inputs, but into a finite dimensional space. In both cases, the expressions of the update equations follow exactly the expressions originally conceived in the linear-in-input context. However, those expressions cannot be directly used in kernel-based filtering due to the infinite dimension of the space where the parameter signal-vector lies, calling for alternative ways of computing the output signal. Fortunately, this task is greatly simplified

thanks to the reproducing property inherent to RKHS. Nonetheless, one must, in principle, store all incoming data inputs or otherwise implement a quantization rule to artificially limit the storage requirements, which can hinder the use of kernel-based filtering in online applications. On the other hand, Volterra-based filtering does not have these drawbacks, for it works in finite dimensional spaces, as linear-in-input filters do.

Regarding the application of Volterra-based SMAF for system identification, the input vector  $\mathbf{x}_N[k]$  is employed so as to produce an output  $y[k] = \mathbf{w}^H[k]\mathbf{x}_N[k]$ . Then, the output error is defined as in the linear-in-input case, but here using the output produced by the Volterra filter.

As for channel equalization, the Volterra filter is employed in the same way as the system identification case, representing a Volterra *feedforward* equalization. Another approach is to employ the Volterra *decision-feedback equalization (DFE)*, illustrated by Figure 3.4, where the Volterra series may be employed in the inputs of both feedforward and feedback filters (as shown in the figure), or separately in each filter input. The adaptive Volterra decision-feedback equalizer works as follows: during its training period, the  $p^{\text{th}}$ -order Volterra input vector  $\mathbf{x}_{\text{FF}_p}[k]$  is assembled by using the nonlinear operator  $V_p(\cdot)$ , whose input is  $\mathbf{x}[k]$ , as shown in Figure 3.4. Then,  $\mathbf{x}_{\text{FF}_p}[k]$  is filtered by the feedforward filter with parameter vector  $\mathbf{w}_{\text{FF}}[k]$ . By using  $V_q(\cdot)$  over the pilot sequence, delayed by  $D$  samples, one is able to build the  $q^{\text{th}}$ -order Volterra input vector of the feedback filter with parameter vector  $\mathbf{w}_{\text{FB}}[k]$ . Therefore, the output of the DFE can be expressed as

$$y[k] = \mathbf{w}_{\text{FF}}^H[k]\mathbf{x}_{\text{FF}_p}[k] + \mathbf{w}_{\text{FB}}^H[k]\mathbf{x}_{\text{FB}_q}[k]. \quad (3.24)$$

Thus, defining  $\mathbf{x}_{\text{DFE}}[k] = [\mathbf{x}_{\text{FF}_p}^T[k] \quad \mathbf{x}_{\text{FB}_q}^T[k]]^T$  and  $\mathbf{w}_{\text{DFE}}[k] = [\mathbf{w}_{\text{FF}}^T[k] \quad \mathbf{w}_{\text{FB}}^T[k]]^T$ , (3.24) can be described in a more compact fashion as

$$y[k] = \mathbf{w}_{\text{DFE}}^H[k]\mathbf{x}_{\text{DFE}}[k]. \quad (3.25)$$

It is worth mentioning that, after the training period, the selector in Figure 3.4 switches to the detected symbol  $\check{y}[k]$ , which is used to assemble the feedback filter input vector  $\mathbf{x}_{\text{FB}_q}[k]$ .

Regarding the applications mentioned above, when using Volterra series, the adaptive filter parameters are accessible, which might be useful to better understand/study the underlying system. For instance, it is possible to analyze the parameters' sensitivity with respect to the inputs, or readily verify the model sparsity. In kernel-based filtering, on the other hand, only the adaptive filter output is observable and the parameter vector cannot be analyzed directly.

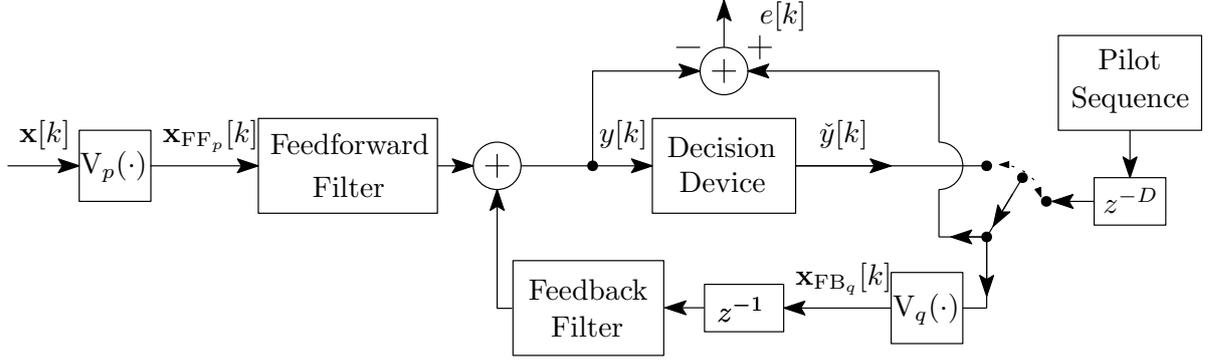


Figure 3.4: Adaptive Volterra decision-feedback equalizer.

### 3.4 Performance Evaluation

This section compares the performance of the proposed Volterra SMAF framework using the algorithms described in Subsection 3.2.1 (SM-PNLMS), and Subsection 3.2.2 (M-BEACON) with the Volterra PNLMS algorithm proposed in [99],<sup>1</sup> and with the Volterra RLS. Although the focus of this work is to present equalization techniques for VLC systems, the methods described in this chapter are suitable for a wide range of nonlinear systems. Therefore, the point of this section is to showcase the performance of the proposed techniques in other setups. The performance evaluation of the proposed filters within the specific VLC context will be conducted in Chapter 6.

Regarding the notation of the techniques presented in this chapter, Volterra PNLMS, Volterra SM-PNLMS, Volterra RLS, and Volterra Modified BEACON algorithms are respectively denoted by V-PNLMS, VSM-PNLMS, V-RLS, and VM-BEACON. Prior simulations indicated that V-PNLMS outperformed Volterra LMS and NLMS algorithms in the tested scenarios, since proportionate techniques are more suitable to work with sparse systems, as those employed in this work and many other Volterra models. It is worth highlighting that VM-BEACON and RLS algorithms present the best performance results, however, their computational complexity per update is quadratic with respect to the filter length  $L$ , while in VSM-PNLMS and V-PNLMS cases, it is linear. Therefore, the proposed techniques of this work will be analyzed considering two benchmarks: VM-BEACON vs. V-RLS and VSM-PNLMS vs. V-PNLMS. It is worth mentioning that the techniques proposed in this chapter aim to work with most of nonlinear systems, including a VLC system.

<sup>1</sup>Here the weighting matrix  $\mathbf{G}[k]$  is the same as the one employed in the set-membership case, whose update equation is described in (3.5).

### 3.4.1 Simulation Methodology and Figures of Merit

The simulation methodology consists in using the two distinct adaptive techniques in the applications described in Section 3, viz.:

- Nonlinear system identification: The mean square error (MSE) is assessed during the training process of the adaptive filters, as well as the convergence speed, average update rates, and *misalignment*, as defined in [100].
- Nonlinear channel equalization: After the training period, the adaptive filters play the role as equalizers, so that their respective bit-error rates (BERs) can be assessed. Two equalization techniques are exploited in this work:
  - Feedforward equalization: Employs the Volterra series in the input vector of the feedforward filter.
  - Decision-feedback equalization: Applies the Volterra series in the input vector of the filters considering three different scenarios, namely: (i) feedforward and feedback filters; (ii) only in feedforward filter; and (iii) only in feedback filter. The motivation here is to analyze how the Volterra filter impacts the performance in each equalizer branch.

### 3.4.2 Simulation Setup

The simulations used 1000 independent runs, a signal-to-noise ratio (SNR) of 30 dB, and 4-PAM symbols as input signals in both system identification and channel equalization scenarios. In the case of set-membership algorithms, the error threshold was set to 0.07 for system identification and 0.28 for channel equalization, for those values represent a good trade-off between steady-state MSE and average update rates. For both simulations, a step-size of  $\mu = 0.8$  was used in V-PNLMS algorithm and a forgetting factor  $\lambda = 0.95$  was employed in V-RLS. Regarding proportionate algorithms,  $\kappa$  was set to 0.5. In order to access properly the tracking performance, two nonlinear Volterra channels [96] were used in both simulation scenarios. These channels were obtained through experimental data generated by magneto-inductive heads which exhibits severe nonlinear effects. Mathematically, the channels are described as

Channel 1:

$$y[k] = 0.5x[k] + 3x[k-1] + 5x[k-2] + 0.3x[k]x[k-1] + 1.2x[k-1]x[k-2].$$

Channel 2:

$$y[k] = 2x[k] + 0.2x^2[k] - 0.7x^2[k-1] + 0.3x[k]x[k-1].$$

In the system identification case, the adaptive filters were trained during 2000 iterations, in which the first 1000 iterations used Channel 1, whereas the remaining iterations employed Channel 2. In addition,  $M = N = 2$ , which means that the unknown systems could, in principle, be perfectly modeled. For the channel equalization case, 10000 iterations were used, but here changing the channel at the 5000<sup>th</sup> iteration. Regarding these channels, the delay in samples  $D$  imposed on the training sequence was  $M + 2$  and  $M - 1$  samples, respectively, where  $M$  is the memory size of the feedforward filter.<sup>2</sup> For BER simulations, 10000 symbols were transmitted and 1000 Monte Carlo runs were performed. The order was  $N = 2$ , while the memory size will be further described.

### 3.4.3 Results for System Identification

Figure 3.5 shows the MSE and misalignment for V-PNLMS, VSM-PNLMS, V-RLS, and VM-BEACON techniques. The convergence speed was also evaluated using the following procedure: considering only the learning curve of Channel 1, it was assumed the respective algorithm converged at the first iteration where the MSE falls within the interval centered at the steady-state MSE (time-average MSE from the 500<sup>th</sup> up to the 1000<sup>th</sup> iteration) and of size two times the standard deviation of the steady-state MSE. Therefore, V-PNLMS, VSM-PNLMS, V-RLS, and VM-BEACON converged after 145, 282, 137, and 180 iterations, respectively. Regarding the average update rates, VSM-PNLMS and VM-BEACON updated in 12.88% and 7.50%, respectively.

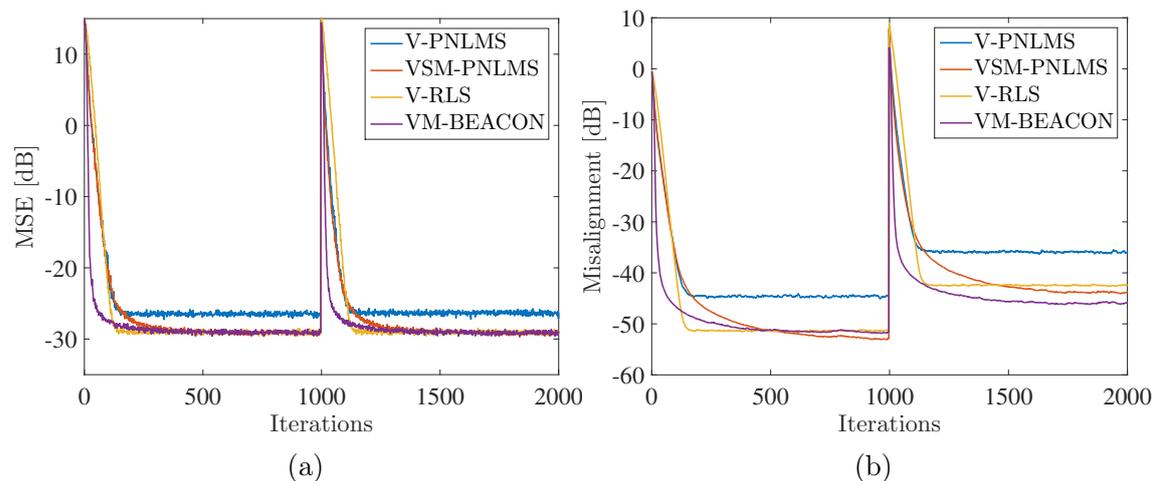


Figure 3.5: MSE (a) and misalignment (b) for nonlinear system identification.

<sup>2</sup>The delay values were chosen based on prior simulations.

## Conclusions of System Identification Simulations

The results presented in Figure 3.5 show that the proposed filters yield better results than their standard counterparts. With respect to misalignment, VSM-PNLMS and VM-BEACON display the lowest levels, outperforming V-PNLMS, yet updating in less than 13% of the iterations in the worst case, showing that the proposed techniques yield better results with much less computational effort.

### 3.4.4 Results for Channel Equalization

Regarding feedforward equalization, Figure 3.6 shows the MSE curves considering the feedforward filter memory  $M_{\text{FF}} \in \{3, 4\}$ , while Figure 3.7 presents the BER results. As in the system identification case, VM-BEACON and V-RLS reach the lowest steady-state MSEs for both  $M_{\text{FF}}$  values, considering Channel 1. In addition, the steady-state MSE of VSM-PNLMS is slightly above VM-BEACON's and V-RLS's, but lower than V-PNLMS's. It is worth mentioning that for  $M_{\text{FF}}$  lower than 3 the adaptive filters were not capable of equalizing Channel 1 and presented MSE levels above 10 dB. Table 3.1 illustrates the results for convergence speed, i.e., how many iterations the respective algorithm needed to reach steady state, and results for average update rates of the algorithms. The procedure of evaluating the convergence speed is the same as in Subsection 3.4.3, but now considering the steady-state MSE from the 4000<sup>th</sup> up to the 5000<sup>th</sup> iteration. Regarding the average update rates of the algorithms, Table 3.1 shows the percentage of iterations in which an update in the filter coefficients was necessary, i.e., when  $\mathbf{w}[k] \notin \mathcal{H}_{\text{Volterra}}[k]$ .

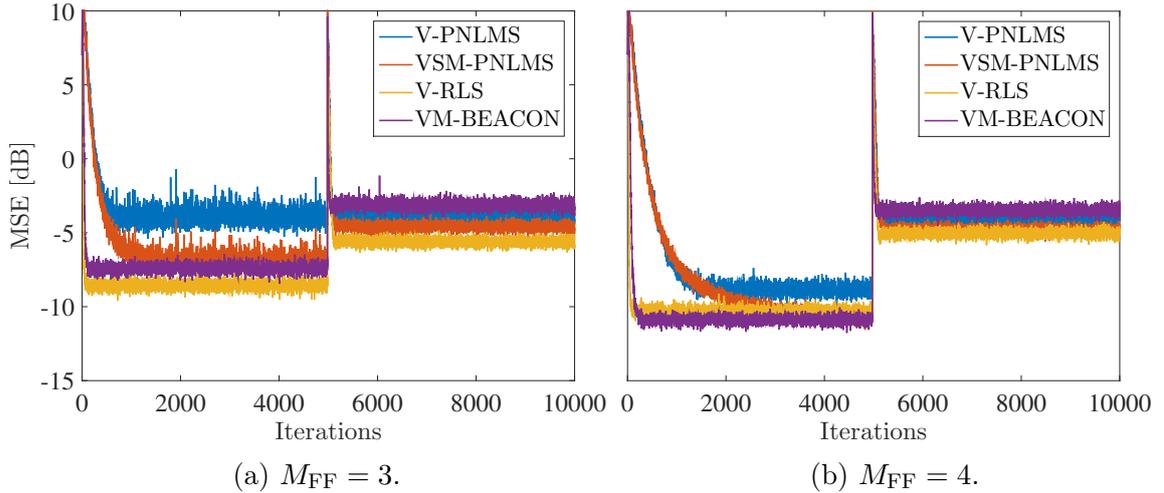


Figure 3.6: MSE for different memory sizes in the feedforward equalization case.

The following results explore the impact of the memory size of feedforward and feedback filters in the DFE case. Considering the VSM-PNLMS algorithm, Figure 3.8 exhibits the average MSEs from the 4000<sup>th</sup> up to the 5000<sup>th</sup> iteration em-

Table 3.1: Iterations until steady state / Average update rates for feedforward equalization.

	V-PNLMS	VSM-PNLMS	V-RLS	VM-BEACON
$M_{\text{FF}} = 3$	422 / 100%	648 / 47.59%	53 / 100%	89 / 46.48%
$M_{\text{FF}} = 4$	1006 / 100%	2140 / 40.18%	58 / 100%	171 / 30.58%

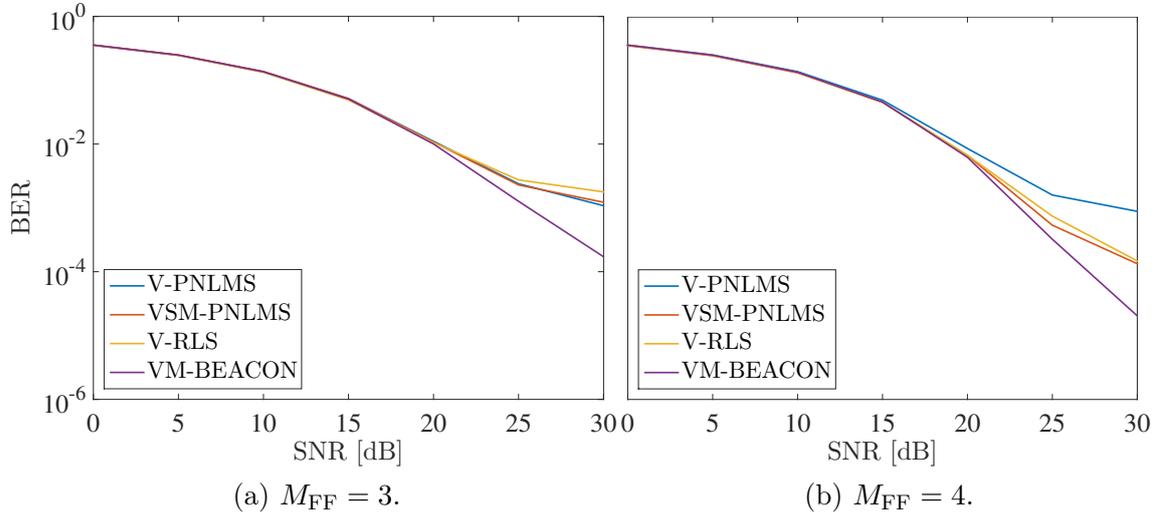


Figure 3.7: BER for different memory sizes in the feedforward equalization case.

employing the Volterra series in both filters (DFE FF & FB in Figure 3.8a), only in the feedforward filter (DFE FF in Figure 3.8b), and only in the feedback filter (DFE FB in Figure 3.8c). Similarly, Figure 3.9 shows the results for the VM-BEACON algorithm. As can be noted from Figs. 3.8 and 3.9, the MSE is more sensitive to the feedforward than the feedback memory size. For instance, in Figure 3.8b, if one considers  $M_{\text{FF}} = 4$ , the steady-state MSE using  $M_{\text{FB}} = 0$  is very close to the one using the  $M_{\text{FB}} = 4$ . It is important to point out that the DFE FF reaches very similar values of steady-state MSE to the DFE FF & FB case, yet using a linear-in-input feedback filter. In the case of DFE FB, the steady-state MSE level is higher than in the other DFE cases. The smallest memory sizes that achieve the lowest steady-state MSE levels are  $M_{\text{FF}} = 4$  and  $M_{\text{FB}} = 0$ . These parameter values are used in the forthcoming simulations to assess the MSE and BER performances employing the DFE FF structure.

Once again, VM-BEACON and V-RLS achieve the best performance, followed by VSM-PNLMS, as illustrated by Figure 3.10. The average update rates are 27.78% and 39.74%, for VM-BEACON and VSM-PLNLS, respectively. Besides, V-PNLMS, VSM-PNLMS, V-RLS, and VM-BEACON reached the steady state after 2010, 2089, 55, and 177 iterations, respectively.

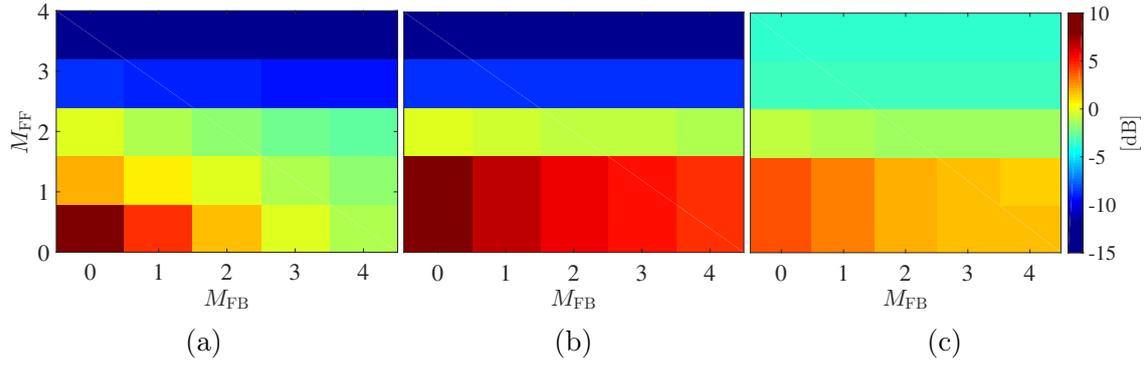


Figure 3.8: Steady-state MSE for different memory sizes for VSM-PNLMS.

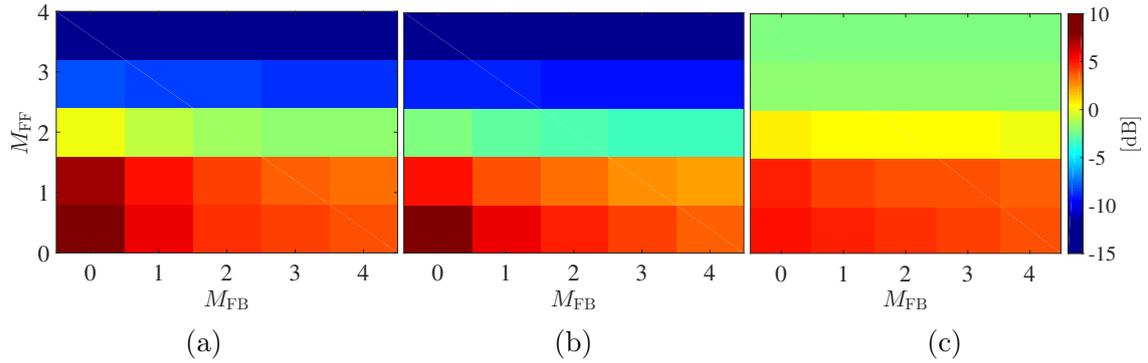


Figure 3.9: Steady-state MSE for different memory sizes for VM-BEACON.

## Conclusions of Equalization Simulations

Regarding the feedforward equalization MSE results, the proposed techniques present lower or very similar steady-state MSE as compared to the competing techniques. VM-BEACON and V-RLS achieved the fastest convergence followed by VSM-PNLMS, but the proposed techniques update in only a fraction of the iterations. It is worth mentioning that VM-BEACON's average update rate is lower than VSM-PNLMS's, but one should note that the computational complexity of the latter is also lower. With respect to BER results, as indicated by the previous MSE plots, the proposed filters present very similar or even better results than the benchmark filters, while also reducing the computational burden.

DFE equalization results show that the MSE is more sensitive to feedforward than feedback memory size. Moreover, using Volterra series only in the feedforward filter yields similar results to the case of using Volterra series in both DFE filters, allowing for computational savings. In addition, using a feedback filter length of 1 yields similar results to the case of employing greater lengths, which indicates that shorter filters should be employed, while larger lengths should be employed in the feedforward filters. Moreover, using Volterra series only in the feedback filter leads to the worst results. By choosing the DFE FF technique and fixing feedforward and feedback memory sizes, VM-BEACON and V-RLS achieve the best performance

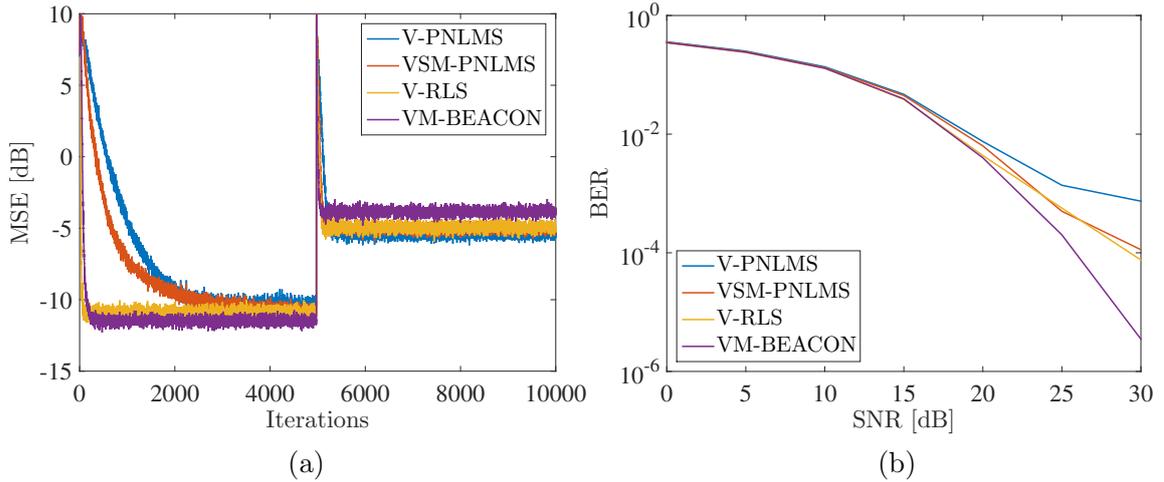


Figure 3.10: MSE (a) and BER (b) for  $M_{\text{FF}} = 4$  and  $M_{\text{FB}} = 0$  in the DFE case.

with respect to MSE and BER, followed by VSM-PNLMS; one should note that the proposed filters present comparable results even though they update in only a small fraction of the iterations.

The results of Section 4.4 indicate that the computational burden of Volterra series can be reduced using the set-membership approach. Nonetheless, these technique disregards the fact that it may be useful to use a threshold  $\bar{\gamma}$  for the linear components of  $\mathbf{w}[k]$ , i.e., for the ones which correspond to the first-order kernel of Volterra series, and employ a different threshold for the nonlinear components of  $\mathbf{w}[k]$ , which are related to higher orders kernels of the Volterra series. In addition, in the channel equalization case, it is desirable that the data used to train the adaptive filters are as minimum as possible, so as to save bandwidth and increase the data throughput of the system. The aforementioned topics will be presented and discussed in Chapter 4.

# Chapter 4

## Semi-Blind Data-Selective and $C$ -Threshold Volterra Adaptive Filtering

The computational burden associated with the use of Volterra series can be quite reduced if the data-selective approach proposed in Chapter 3 is employed. Nevertheless, there is still room for improvements, specially considering two practical requirements in communication system: spectral efficiency and low computational complexity (power-limited mobile devices impose constraints on computational burden). Therefore, this chapter focuses on tailoring the contributions of Chapter 3 to meet these demands.

One of the main drawbacks of Volterra series is the computational burden given the large number of parameters to be updated. In addition, during the training of adaptive filters that employ such series, the speed of convergence is often slow, requiring a considerable amount of training data, thereby, reducing the spectral efficiency. In order to address this issue, one can employ semi-blind data-selective techniques, which replace the reference signal with a decided version of the output of the adaptive filter. Through this chapter, these methods are used along with the techniques presented in Chapter 3.

This chapter also proposes a technique that takes advantage of the structure of the Volterra series to reduce even more its computational burden. The idea is to employ a combination of adaptive filters along with partial update algorithms in order to split the update process according to the kernels of the Volterra series. Higher order kernels present more elements than lower kernels. Therefore, if one assigns larger error thresholds to the higher order kernels, the update rates of the corresponding adaptive filters decrease, reducing the overall computational complexity.

## 4.1 Semi-Blind Volterra-based SMAF

During the training of the adaptive filters, a reference signal is sent by the transmitter to the receiver until these filters reach their convergence. Nonetheless, depending on the length of these filters, the speed of convergence may not be so fast, which would require a large amount of data containing reference signal, decreasing the spectral efficiency. In the case of Volterra adaptive filtering, as already mentioned, the long length of the parameter vector is one of the drawbacks of this method. Therefore, a good approach to enhance the spectral efficiency when using Volterra series is to connect the semi-blind framework to the data-selective Volterra adaptive filtering scheme proposed in Chapter 3.

### 4.1.1 VSBSM-PNLMS

The semi-blind SM-PNLMS (SBSM-PNLMS) for quadrature amplitude modulation (QAM) was first derived in [76], aiming at achieving a compromise among complexity, speed of convergence, and spectral efficiency. Classic blind equalizers, such as the family of constant modulus algorithms (CMA), may converge to local minimum and present slow convergence [68]. Besides, as the SBSM-PNLMS technique employs a data-selective scheme, this method presents lower computational complexity if compared to popular algorithms, e.g., NLMS. The authors in [76] show that SBSM-PNLMS algorithm is a generalization of the SM-PNLMS. Connecting the ideas in [76] with the ideas of Chapter 3, we propose the Volterra SBSM-PNLMS (VSBSM-PNLMS) filter, whose coefficients are updated according to a modified version of (3.4), as follows

$$\begin{aligned} \mathbf{w}[k+1] = \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \\ \text{subject to : } \|d[k] - \mathbf{w}^H \mathbf{x}_N[k] - \bar{\gamma} \text{sign}_{\infty}\{e[k]\}\|_{\infty} = 0, \end{aligned} \quad (4.1)$$

where  $\text{sign}_{\infty}\{e[k]\}$  is defined as

$$\text{sign}_{\infty}\{e[k]\} = \begin{cases} \frac{e[k]}{\|e[k]\|_{\infty}}, & \text{if } e[k] \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (4.2)$$

where  $\|\cdot\|_{\infty}$  stands for the infinite-norm. It is worth mentioning that when working with complex values, they are here interpreted as points in  $\mathbb{R}^2$ . Thus, the formulation

considers that, if  $z = z_{\text{real}} + jz_{\text{imag}} \in \mathbb{C}$ , then

$$\begin{aligned} \|z\|_{\infty} &= \|z_{\text{real}} + jz_{\text{imag}}\|_{\infty} \\ &= \|(z_{\text{real}}, z_{\text{imag}})\|_{\infty}, \\ &= \max\{|z_{\text{real}}|, |z_{\text{imag}}|\}. \end{aligned} \quad (4.3)$$

In the unsupervised period, the only adjustment to be made is to replace  $d[k]$  with  $b[k] = \text{dec}\{y[k]\}$ , where  $\text{dec}\{\cdot\}$  stands for a symbol decision operator. The update equations related to (4.1) are derived in Appendix A.4.

The authors in [76] proposed the following update rules: during the supervised period, an update is necessary if the equalized symbol  $y[k] = \mathbf{w}^H[k]\mathbf{x}_N[k]$  is out of the hard-decision region of the employed constellation, e.g., quadrature and pulse amplitude modulations, and if, in the unsupervised period,  $y[k]$  is out of the aforementioned hard-decision region. One should note, that in the latter case, the reference signal is  $b[k] = \text{dec}\{y[k]\}$ . Regarding the hard-decision region, strictly speaking, it represents a neighborhood of a valid symbol for the reference signal. Considering the supervised and unsupervised periods, an update is required if

$$\|d[k] - y[k]\|_{\infty} > \bar{\gamma} \quad (\text{Supervised period}), \quad (4.4)$$

$$\|b[k] - y[k]\|_{\infty} > \bar{\gamma} \quad (\text{Unsupervised period}). \quad (4.5)$$

Thus, defining  $e_b[k] = b[k] - y[k]$ , one can modify (3.6) as follows:

$$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{\|e[k]\|_{\infty}}, & \text{if } \|e[k]\|_{\infty} > \bar{\gamma}, \\ 1 - \frac{\bar{\gamma}}{\|e_b[k]\|_{\infty}}, & \text{if } \|e_b[k]\|_{\infty} > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (4.6)$$

Hence,

$$\mathbf{w}[k+1] = \begin{cases} \mathbf{w}[k] + \mu[k] \frac{\mathbf{G}[k]\mathbf{x}_N[k]}{\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]} e^*[k] & (\text{Supervised period}), \\ \mathbf{w}[k] + \mu[k] \frac{\mathbf{G}[k]\mathbf{x}_N[k]}{\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]} e_b^*[k] & (\text{Unsupervised period}). \end{cases} \quad (4.7)$$

It is worth mentioning that the generalization for the affine projection algorithm can be found in [76]. For better understanding, the pseudo-code of this technique is described in Appendix B.3.

### 4.1.2 VSB Modified BEACON

The M-BEACON algorithm can also be adjusted to work in a semi-blind fashion. As in the VSBSM-PNLMS's case, the infinite-norm takes place on the constraint of

the cost function of this technique, which is described by modifying (3.9) as

$$\begin{aligned} \mathbf{w}[k] = \arg \min_{\mathbf{w}} \cdot & \left( \sum_{i=1}^k \lambda^{k-i}[k] |d[i] - \mathbf{w}^H \mathbf{x}[i]|^2 \right) \\ \text{subject to : } & \|d[k] - \mathbf{w}^H \mathbf{x}[k]\|^2 - \bar{\gamma}^2 = 0. \end{aligned} \quad (4.8)$$

Then, one can express the update equations for this technique as

$$\lambda[k] = \begin{cases} \mathbf{x}_N^H[k] \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}_N[k] \frac{1}{\left(\frac{\|e[k]\|_\infty}{\bar{\gamma}} - 1\right)}, & \text{if } \|e[k]\|_\infty > \bar{\gamma}, \\ \mathbf{x}_N^H[k] \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}_N[k] \frac{1}{\left(\frac{\|e_b[k]\|_\infty}{\bar{\gamma}} - 1\right)}, & \text{if } \|e_b[k]\|_\infty > \bar{\gamma}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.9)$$

yielding the update equations during supervised and unsupervised periods as

$$\mathbf{w}[k] = \begin{cases} \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}_N[k] e^*[k], & \text{(Supervised period),} \\ \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}_N[k] e_b^*[k]. & \text{(Unsupervised period).} \end{cases} \quad (4.10)$$

The resulting filter will be named Volterra semi-blind M-BEACON (VSBM-BEACON). One should note that the *a priori* error during the supervised period is defined here as  $e[k] = \mathbf{w}^H[k-1] \mathbf{x}[k]$ . Hence, in the unsupervised period, as the reference signal  $d[k]$  is replaced by  $b[k]$ , the error  $e_b[k]$  is defined as  $e_b[k] = b[k] - \mathbf{w}^H[k-1] \mathbf{x}[k]$ . In Appendix B.4, the pseudo-code of this technique is detailed.

### 4.1.3 From Supervised to Blind Training

A question that may rise is how to define the iteration in which the decided output of the adaptive filter will play the role as reference signal. In case of an early switch, i.e., the unsupervised period starts when the adaptive filter is far from its convergence, it may interrupt the learning of the filter and lead to a high steady-state MSE level. On the other hand, if a late switch occurs, a large quantity of reference signal would be unnecessarily employed. In order to find a good compromise between steady-state MSE level and amount of reference signal, it is proposed a rule based on the sample median of  $\mathbf{e}_{\text{SB}} = [|e[k-1]| |e[k-2]| \cdots |e[k-L_{\text{SB}}-1]|]^T$ ,  $L_{\text{SB}} \in \mathbb{N}$ . Therefore, the unsupervised period starts at the first iteration that satisfies

$$\mathcal{M}[\mathbf{e}_{\text{SB}}] < \bar{\gamma}_{\text{SB}}, \quad (4.11)$$

where  $\mathcal{M}[\cdot]$  represents the sample median operator and  $\bar{\gamma}_{\text{SB}} \in \mathbb{R}_+$ . Regarding the threshold  $\bar{\gamma}_{\text{SB}}$ , it is possible to set it according to the constellation employed in the system. Considering a standard 4-PAM constellation, the width of the decision

region is two. Therefore, the aforementioned threshold can be set to a fraction of this width, i.e.,  $\bar{\gamma}_{\text{SB}} = 2\eta$ , where  $\eta \in [0, 1]$  is a threshold that controls the width of the sub-region area within the decision region, as illustrated by Figure 4.1. Thus, if  $\mathcal{M}[\mathbf{e}_{\text{SB}}]$  falls within the regions defined by  $\eta$ , the unsupervised period takes place.

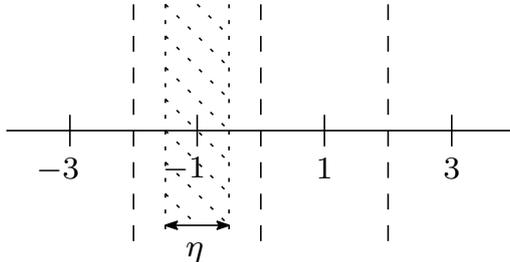


Figure 4.1: 4-PAM decision regions and the subregion defined by  $\eta$ .

## 4.2 C-Threshold Volterra-based SMAF

The results of Chapter 3 indicated that the computational complexity of Volterra series can be remarkably reduced. Nonetheless, these results can be further improved by analyzing the intrinsic structure of the Volterra series. Indeed, it is possible to employ a data-selective adaptive filter to update separately the elements of each kernel of the Volterra series,<sup>1</sup> resulting in  $C \in \mathbb{N}$  adaptive filters with parameter vectors  $\mathbf{w}_c[k] \in \mathbb{R}^L$ ,  $c \in \{1, \dots, C\}$ , and memory size  $M$ , where the  $c^{\text{th}}$  filter only updates  $\binom{M+c}{c}$  elements, via a partial-update approach [68, 101]. Each of these filters update more elements as the order of the kernels increases. Then, by assigning larger error thresholds to these filters, one can reduce their update rates, thus reducing the overall computational complexity. In the end, the parameter vector  $\mathbf{w}[k+1]$  is assembled by combining all  $\mathbf{w}_c[k+1]$ .

Regarding the partial-update scheme, such method is employed when the computational complexity needs to be managed. In applications where a high number of parameters to be updated is involved, such as Volterra filters and echo cancellation, given the computational complexity constraint, a good approach would be to update only part of the parameters at each iteration. The partial-update scheme may be presented considering two branches: (i) fixed partial update, for which a predefined number of parameters is updated at every iteration; (ii) variable partial update, for which the number of parameters can vary. These schemes are performed by using a diagonal matrix  $\mathbf{U}[k] \in \mathbb{N}^{L \times L}$ , whose diagonal elements are “1” or “0”. Thus,

<sup>1</sup>The elements of the kernels are related to the order of the product of the terms of the Volterra series. For example, the second-order kernel is assembled by elements corresponding to the second-order products of the Volterra series, while the third-order kernel is formed by the third-order products, and so on.

the  $l^{\text{th}}$  element of  $\mathbf{w}_c[k]$  is updated at the  $k^{\text{th}}$  iteration if  $\mathbf{U}_l[k] = 1$ . By using this technique in the SM-PNLMS algorithm, it results in the SM-partial-update-PNLMS (SM-PUPNLMS), which updates according to the following cost function:

$$\begin{aligned} \mathbf{w}[k+1] &= \arg \min_{\mathbf{w}} \cdot \|\mathbf{w} - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \\ \text{subject to : } & d[k] - \mathbf{w}^H \mathbf{x}[k] - \bar{\gamma} \text{sign}\{e[k]\} = 0 \\ & \tilde{\mathbf{U}}[k](\mathbf{w} - \mathbf{w}[k]) = \mathbf{0}_L, \end{aligned} \quad (4.12)$$

where  $\tilde{\mathbf{U}}[k] = \mathbf{I} - \mathbf{U}[k]$  represents the complementary matrix of  $\mathbf{U}[k]$ , used to ensure that  $\tilde{\mathbf{U}}[k]\mathbf{w} = \tilde{\mathbf{U}}[k]\mathbf{w}[k]$ , and  $\mathbf{0}_L$  represents a vector of zeros with dimension  $L$ . After solving the optimization problem described in (4.12), one can express the update equations for this technique as

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu[k] \frac{\mathbf{U}[k]\mathbf{G}[k]\mathbf{x}[k]}{\mathbf{x}^H[k]\mathbf{U}[k]\mathbf{G}[k]\mathbf{x}[k]} e^*[k], \quad (4.13)$$

whose update equations for  $\mu[k]$  and  $\mathbf{G}[k]$  are identical to the SM-PNLMS case, described in (3.5) and (3.6), respectively. In addition,  $\mathbf{w}[k]$  is updated in the direction of the selected entries of  $\mathbf{G}[k]\mathbf{x}[k]$ , whose selection is performed by  $\mathbf{U}[k]$ .

With respect to the combination of adaptive filters, their main purpose is to associate the most noticeable characteristics of each filter to reach a desirable effect [74, 75]. For instance, one can combine the parameter vectors of two LMS filters with different step-sizes to reach a lower MSE steady state, even with a fast convergence speed, which could be very hard to achieve using only one LMS filter. Considering  $C$  filters, one can define the output and error of the  $c^{\text{th}}$  filter as

$$y_c[k] = \mathbf{w}_c^H[k]\mathbf{x}_N[k], \quad (4.14)$$

$$e_c[k] = d[k] - y_c[k]. \quad (4.15)$$

Therefore, the overall output  $y[k]$ , the output error  $e[k]$ , and the parameter vector  $\mathbf{w}[k+1]$  are given by

$$y[k] = \sigma_1[k]y_1[k] + \sigma_2[k]y_2[k] + \cdots + \sigma_C[k]y_C[k], \quad (4.16)$$

$$e[k] = \sigma_1[k]e_1[k] + \sigma_2[k]e_2[k] + \cdots + \sigma_C[k]e_C[k], \quad (4.17)$$

$$\mathbf{w}[k+1] = \sigma_1[k]\mathbf{w}_1[k+1] + \sigma_2[k]\mathbf{w}_2[k+1] + \cdots + \sigma_C[k]\mathbf{w}_C[k+1], \quad (4.18)$$

where  $\sigma_c \in \mathbb{R}$  represents the combination parameter. The updating process occurs as follows: first, the parameter vectors  $\mathbf{w}_c[k], \forall c \in \{1, \dots, C\}$ , update according to a given adaptive algorithm, e.g., LMS or PNLMS. Then, these parameter vectors are combined by a combination layer, which ponders the parameter vectors according

to a given rule, thus assembling the parameter vector  $\mathbf{w}[k+1]$ . Regarding the combination schemes, two techniques are the most popular, viz.: convex and affine combinations [74]. Considering the former,  $\sigma_c[k]$  must also satisfy  $\sum_{c=1}^C \sigma_c[k] = 1$ . In the latter case,  $\sigma_c[k]$  is constrained to be any real number.

In addition to the combination of the parameter vectors of the  $C$  adaptive filters, it is possible to perform a weight transfer scheme among all vectors  $\mathbf{w}_c[k]$ , or even a periodical feedback of  $\mathbf{w}[k+1]$  to the other parameter vectors, i.e., after building  $\mathbf{w}[k+1]$ , this vector overwrites  $\mathbf{w}_c[k+1], \forall c \in \{1, \dots, C\}$ , at iterations  $k = k_0K$  for some  $k_0 \in \mathbb{N}$  and for a given  $K \in \mathbb{N}$ . Since this combination scheme better fits with our proposal, this technique will be used in this work; further details can be found in [102, 103]. Therefore, Figure 4.2 illustrates the generalized topology of the combination of  $C$  filters with the feedback process, where the combination layer performs the operation described in (4.16). Then, considering that the  $c^{\text{th}}$  filter updates according to the SM-PNLMS algorithm, one can describe the  $l^{\text{th}}$  diagonal entry of  $\mathbf{G}_c[k]$  as

$$\mathbf{G}_{c,ll}[k] = \frac{1 - \kappa_c \mu_c[k]}{L} + \frac{\kappa_c \mu_c[k] |\mathbf{w}_{c,l}[k]|}{\|\mathbf{w}_c[k]\|_1}, \quad (4.19)$$

where  $\mathbf{w}_{c,l}[k]$  denotes the  $l^{\text{th}}$  entry of  $\mathbf{w}_c[k]$  and

$$\mu_c[k] = \begin{cases} 1 - \frac{\bar{\gamma}_c}{|e_c[k]|}, & \text{if } |e_c[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (4.20)$$

Therefore, the update equation of the  $\mathbf{w}_c[k]$  is described as

$$\mathbf{w}_c[k+1] = \mathbf{w}_c[k] + \mu_c[k] \frac{\mathbf{G}_c[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{G}_c[k] \mathbf{x}_N[k]} e_c^*[k]. \quad (4.21)$$

Nonetheless, if one considers a scheme which continuously feedbacks  $\mathbf{w}[k+1]$ , i.e., for  $K = 1$ , the output of the  $C$  filters are the same at every iteration, as well the output errors. Hence, no update equation for  $\sigma_c[k]$  is required; besides, the evaluation of the errors  $e_c[k]$  are not necessary. Then, by connecting the partial-update scheme to the idea of combinations of filters, one can assign a filter for each kernel of the Volterra series, and update its elements separately from the other kernels. Therefore, by assigning the largest error thresholds for those filters which update the kernels of higher orders, it averts unnecessary updates on filters which update the lower order kernels. Indeed, higher order kernels lead to more elements. Therefore, it is more efficient, from the computational burden point of view, to avoid their updates. Thus, the filters assigned to update higher kernels are treated more

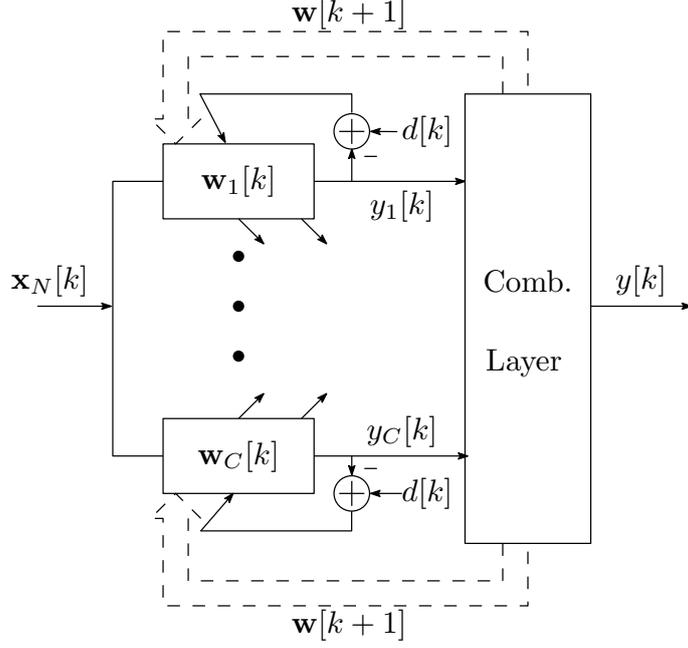


Figure 4.2: Generic topology of the combination of  $C$  adaptive filters.

leniently. It is possible to define the following matrices

$$\begin{aligned}
 \mathbf{U}_1 &= \text{diag}([\mathbf{1}_{M+1}^T \mathbf{0}_{L-M-1}^T]), \\
 \mathbf{U}_2 &= \text{diag}([\mathbf{0}_{M+1}^T \mathbf{1}_{L_2}^T \mathbf{0}_{L-L_2-M-1}^T]), \\
 &\vdots \\
 \mathbf{U}_{C-1} &= \text{diag}([\mathbf{0}_{L-L_{N-1}-L_N}^T \mathbf{1}_{L_{N-1}}^T \mathbf{0}_{L_N}^T]), \\
 \mathbf{U}_C &= \text{diag}([\mathbf{0}_{L-L_N}^T \mathbf{1}_{L_N}^T]), \tag{4.22}
 \end{aligned}$$

where  $L$  denotes the total length of the parameter vector, as described in (3.21), and  $L_N = \binom{N+M}{N}$ . Then,  $\mathbf{w}_c[k]$  is updated as

$$\mathbf{w}_c[k+1] = \mathbf{w}_c[k] + \mu_c[k] \frac{\mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]} e^*[k]. \tag{4.23}$$

By using (4.18), one can assemble the vector  $\mathbf{w}[k+1]$ . Since the focus of this work is to propose nonlinear equalizers for VLC systems, rarely an order of the Volterra series higher than two is needed. Therefore, we restrict the use of the  $C$ -threshold technique to the second-order Volterra series. Hence, this technique will be named Volterra double-threshold SM-PNLMS (VDTSM-PNLMS). Nevertheless, the proposed method is suitable for other types of nonlinear systems, which may demand higher orders. For the sake of simplicity, the  $C$ -threshold technique is presented in a decision-feedback equalization scenario where the second-order Volterra series is employed in both feedforward and feedback filters. In this case, the elements of  $\mathbf{w}_1[k]$

and  $\mathbf{w}_2[k]$  correspond to the first- and second-order kernels of the Volterra series, respectively. These vectors are here denoted as  $\mathbf{w}_{\text{DFE}_1}[k]$  and  $\mathbf{w}_{\text{DFE}_2}[k]$ , respectively. In the same manner, the matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2$  are denoted as  $\mathbf{U}_{\text{DFE}_1}[k]$  and  $\mathbf{U}_{\text{DFE}_2}[k]$ , and described as

$$\mathbf{U}_{\text{DFE}_1} = \text{diag} \left( [\mathbf{1}_{M_{\text{FF}}+1}^T \mathbf{0}_{L_{\text{FF}}-M_{\text{FF}}-1}^T \mathbf{1}_{M_{\text{FB}}+1}^T \mathbf{0}_{L_{\text{FB}}-M_{\text{FB}}-1}^T]^T \right), \quad (4.24)$$

$$\mathbf{U}_{\text{DFE}_2} = \text{diag} \left( [\mathbf{0}_{M_{\text{FF}}+1}^T \mathbf{1}_{L_{\text{FF}}-M_{\text{FF}}-1}^T \mathbf{0}_{M_{\text{FB}}+1}^T \mathbf{1}_{L_{\text{FB}}-M_{\text{FB}}-1}^T]^T \right), \quad (4.25)$$

where  $L_{\text{FF}} = \frac{(M_{\text{FF}}+1+2)!}{(M_{\text{FF}}+1)!} - 1$  represents the length of the feedforward parameter vector with memory-size of  $M_{\text{FF}}$ . Analogously,  $L_{\text{FB}}$  denotes the length of the feedback parameter vector, with  $M_{\text{FB}}$  being the memory-size of the feedback counterpart. Since the equations for  $\mathbf{U}_{\text{DFE}_1}$  and  $\mathbf{U}_{\text{DFE}_2}$  are presented, by using  $\mathbf{x}_{\text{DFE}}[k]$  as input vector (see Section 3.3 for the definition of  $\mathbf{x}_{\text{DFE}}[k]$ ), one can update  $\mathbf{w}_c[k]$  as follows:

$$\mathbf{w}_c[k+1] = \mathbf{w}_c[k] + \mu_c[k] \frac{\mathbf{U}_{\text{DFE}_c} \mathbf{G}_c[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{U}_{\text{DFE}_c} \mathbf{G}_c[k] \mathbf{x}_N[k]} e^*[k]. \quad (4.26)$$

Therefore, by assuming a convex combination scheme, one can rewrite (4.18) for the case of two filters as

$$\mathbf{w}[k+1] = \sigma \mathbf{w}_1[k+1] + (1 - \sigma) \mathbf{w}_2[k+1], \quad (4.27)$$

where  $\sigma \in \{0, 1\}$  is the convex combination parameter.

### 4.2.1 Computational Efficiency of Volterra Double-Threshold Algorithm

For the sake of simplicity, consider again a second-order Volterra filter. Unlike the Volterra SM-PNLMS technique proposed in Chapter 3, we need now two parameters to characterize the update rates related to the double-threshold method, since the parameter vectors  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  may update separately. Therefore, it lacks a figure of merit capable of comparing the computational complexity of these techniques. Then, in this subsection, a figure of merit which considers the number of updated elements that each technique provides is described, as well as the condition that must be satisfied in order to state which technique is more efficient.

Considering  $\bar{\gamma}_1 < \bar{\gamma}_2$ , if  $e[k] > \bar{\gamma}_2$ , both  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  are updated. Hence,  $M+1$  elements are updated in  $\mathbf{w}_1[k]$ , and  $\binom{M+2}{2}$  elements are updated in  $\mathbf{w}_2[k]$ , totaling  $L$  elements.<sup>2</sup> In the case of  $\bar{\gamma}_1 < e[k] < \bar{\gamma}_2$ , i.e., when only  $\mathbf{w}_1[k]$  updates,

<sup>2</sup>Despite the sum of number of elements of  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  is  $2L$ , due to the partial-update technique, only  $L$  elements are actually updated in this case.

just  $L_1 = M + 1$  elements are updated. Therefore, considering double-threshold algorithms, and defining as  $\nu_1$  and  $\nu_2$  the update rates of  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$ , respectively, one can express the average number of updated elements (NUE) per run of training of the adaptive filter as

$$\text{NUE}_{\text{DT}} = \frac{(\nu_1 L_1 + \nu_2 L) N_{\text{it}}}{100}, \quad (4.28)$$

where  $N_{\text{it}}$  represents the number of iterations. In the case of the VSM-PNLMS technique, the average NUE can be written as

$$\text{NUE} = \frac{\nu \times L \times N_{\text{it}}}{100}, \quad (4.29)$$

where  $\nu$  stands for the update rate of standard Volterra-based algorithms. Then, the proposed double-threshold algorithm is more efficient than VSM-PNLMS if  $\text{NUE}_{\text{DT}} < \text{NUE}$ , i.e.,

$$\frac{(\nu_1 L_1 + \nu_2 L) N_{\text{it}}}{100} < \frac{\nu \times L \times N_{\text{it}}}{100}$$

$$\nu_1(M + 1) + \nu_2 \left[ \frac{(M + 1)^2 + 3(M + 1)}{2} \right] < \nu \left[ \frac{(M + 1)^2 + 3(M + 1)}{2} \right], \quad (4.30)$$

$$(4.31)$$

leading to

$$M > \frac{2\nu_1}{\nu - \nu_2} - 4, \text{ for } \nu > \nu_2. \quad (4.32)$$

Considering  $\nu \leq \nu_2$ ,  $\text{NUE}_{\text{DT}}$  is always larger than NUE for any value of  $\nu_1$ . Hence, VDTSM-PNLMS is less efficient in this case.

In the DFE case, due to the memory of the feedback filter, the length  $L_1$  increases to  $L_1 = M_{\text{FF}} + 1 + M_{\text{FB}} + 1$ , where  $M_{\text{FF}}$  and  $M_{\text{FB}}$  stand for the feedforward and feedback memories, respectively. In addition,  $L$  depends on which filters the Volterra series is employed. Hence, the three possible values for  $L$  are

1. Volterra series in feedforward and feedback filters:  $L = \frac{(M_{\text{FF}}+1)^2+3(M_{\text{FF}}+1)}{2} + \frac{(M_{\text{FB}}+1)^2+3(M_{\text{FB}}+1)}{2}$ ,
2. Volterra series only in feedforward filter:  $L = \frac{(M_{\text{FF}}+1)^2+3(M_{\text{FF}}+1)}{2} + M_{\text{FB}} + 1$ ,
3. Volterra series only in feedback filter:  $L = \frac{(M_{\text{FB}}+1)^2+3(M_{\text{FB}}+1)}{2} + M_{\text{FF}} + 1$ .

Therefore, the proposed Volterra double-threshold algorithm is more efficient if (4.30) is satisfied.

### 4.3 Semi-blind $C$ -Threshold Volterra-based SMAF

It is also possible to join the  $C$ -threshold technique described in Section 4.2 with the one presented in Section 4.1 to achieve further reduction of computational complexity and higher spectral efficiency. Therefore, one can write the update equations of Volterra semi-blind double-threshold SM-PNLMS (VSBDTSM-PNLMS) method by modifying (4.23) and (4.7) as

$$\mathbf{w}_c[k+1] = \begin{cases} \mathbf{w}_c[k] + \mu_c[k] \frac{\mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]} e_c^*[k] & (\text{Supervised period}), \\ \mathbf{w}_c[k] + \mu_c[k] \frac{\mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k]} e_{c_b}^*[k] & (\text{Unsupervised period}), \end{cases} \quad (4.33)$$

where

$$\mu_c[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{\|e_c[k]\|_\infty}, & \text{if } \|e_c[k]\|_\infty > \bar{\gamma}_c, \\ 1 - \frac{\bar{\gamma}}{\|e_{c_b}[k]\|_\infty}, & \text{if } \|e_{c_b}[k]\|_\infty > \bar{\gamma}_c, \\ 0, & \text{otherwise,} \end{cases} \quad (4.34)$$

with  $e_{c_b}[k]$  representing the output error of the  $c^{\text{th}}$  filter during the unsupervised period. Then,  $\mathbf{w}[k+1]$  is assessed by using (4.27). After that, this vector is fed back to  $\mathbf{w}_c[k+1]$ ,  $c \in \{1, \dots, C\}$ . Note that these steps are identical to the ones presented in Section 4.2, with the difference that now one must consider the use of the infinite-norm over  $e_c[k]$  or  $e_{c_b}[k]$  in order to form  $\mu_c[k]$  and  $\mathbf{w}_c[k+1]$ . In order to clarify the implementation of this technique, Appendix B.6 describes it in the form of pseudo-code.

### 4.4 Performance Evaluation

This section assesses the performance of the proposed techniques described in Section 4.2 and Section 4.1. Considering for all algorithms presented here the use of the second-order Volterra series, the VDTSM-PNLMS algorithm will be compared with the VSM-PNLMS technique in both system identification and equalization scenarios. After that, the results considering the techniques described in Subsection 4.1.1 (VSBSM-PNLMS), Subsection 4.1.1 (VSBM-BEACON), and Section 4.3 (VDTSSM-PNLMS) will be presented. As in Chapter 3, the techniques proposed in this chapter are suitable for a wide range of nonlinear systems. Therefore, the results described here show the performance of the techniques in a generic fashion. Chapter 6 contains results for a VLC scenario.

### 4.4.1 Simulation Methodology and Figures of Merit

The simulation methodology of this chapter is similar to the one presented in Subsection 3.4.1. Regarding the figures of merit, the main differences are the ones included to analyze the results of the techniques proposed in this chapter, viz.:

- Double-threshold algorithms: MSE and BER are assessed using different values of error threshold  $\bar{\gamma}_2$ . Moreover, as the filters  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  may update separately, their update rates are also appraised independently. Hence, the update rates of  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  are denoted as  $\nu_1$  and  $\nu_2$ , respectively. In addition, the NUE of each filter is displayed, in order to correctly analyze the computational complexity.
- Semi-blind algorithms: MSE and BER are evaluated for distinct values of  $\eta$ . Besides, the average iteration in which the unsupervised period starts is also assessed (named *Blind iteration*), as well as the update rate during supervised and unsupervised periods, represented by  $\nu_{\text{SP}}$  and  $\nu_{\text{BP}}$ , and the overall update rate. Regarding the *Blind iteration*, it represents the average iteration in which the unsupervised period started, considering all independent runs, which also indicates the number of symbols employed to train the adaptive filters, in average; one should note, however, that the average iteration is evaluated disregarding the runs in which the unsupervised period could not be reached, i.e., when the *Blind iteration* was not defined.
- Semi-blind double-threshold algorithm: MSE and BER are evaluated for distinct values of threshold  $\eta$  and  $\bar{\gamma}_2$ . As in the other semi-blind techniques, the *Blind iteration* is also assessed, as well as the update rates for both  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$  during supervised and unsupervised periods, represented by  $\nu_{\text{SP}_1}$ ,  $\nu_{\text{BP}_1}$ ,  $\nu_{\text{SP}_2}$ , and  $\nu_{\text{BP}_2}$ .

Table 4.1 summarizes the parameters of the techniques used in the simulations, as well as figures of merit assessed from the results.

### 4.4.2 Simulation Setup

The simulation setup employed here is quite similar to the one presented in Subsection 3.4.2. Considering the semi-blind techniques, the nonlinear system employed was Channel 1 (see 3.4.2), since all the results described in Chapter 3 are based on it. In addition, the only purpose of Channel 2 is to assess the tracking performance of Volterra-based filters, which was thoroughly performed in the preceding chapter. Regarding proportionate algorithms,  $\kappa_1$  and  $\kappa_2$  were set to 0.5. Considering double-threshold algorithms, the error threshold  $\bar{\gamma}_1$  was set to 0.07 for system identification

Table 4.1: Description of parameters and figures of merit.

	Description
$\bar{\gamma}_1$	Error threshold for $\mathbf{w}_1[k]$
$\bar{\gamma}_2$	Error threshold for $\mathbf{w}_2[k]$
$\eta$	Threshold used to indicate when unsupervised period starts
<i>Blind iteration</i>	Mean iteration in which the unsupervised period started
$\nu_{\text{SP}_1}$	Update rate of $\mathbf{w}_1[k]$ during supervised period
$\nu_{\text{SP}_2}$	Update rate of $\mathbf{w}_2[k]$ during supervised period
$\nu_{\text{BP}_1}$	Update rate of $\mathbf{w}_1[k]$ during unsupervised period
$\nu_{\text{BP}_2}$	Update rate of $\mathbf{w}_2[k]$ during unsupervised period

and 0.28 for channel equalization, whereas  $\bar{\gamma}_2$  was set from  $\bar{\gamma}_2 = \bar{\gamma}_1$  to  $\bar{\gamma}_2 = 4\bar{\gamma}_1$  using an interval of 0.5. This variation of  $\bar{\gamma}_2$  was performed in order to better understand the relation between update rate as a function of the value of  $\bar{\gamma}_2$ , as well as the resulting steady-state MSE and BER. Besides, the combination parameter  $\sigma$  was chosen as 0.5.

Regarding semi-blind and semi-blind double-threshold algorithms, the feedforward memory  $M_{\text{FF}}$  was increased to 8 in order to analyze properly the semi-blind characteristics of the techniques. Prior simulations indicated that when a memory-size in the feedforward filter of 4 was employed (as in the results of Chapter 3 and in the results of double-threshold-based algorithms of this chapter), the MSE curve of semi-blind algorithms presented a forgetting effect, i.e., after reaching a low steady-state MSE level, this MSE level monotonically increased. It is worth highlighting that the delay imposed during the training of the adaptive filters were the same employed in Subsection 3.4.2.

### 4.4.3 Results for System Identification

Figure 4.3 shows the MSE and misalignment for VDTSM-PNLMS algorithm. The figure of merit chosen to define when the adaptive filter had converged is the same employed in Subsection 3.4.3, viz.: considering only Channel 1, the algorithm converges at the first iteration where the MSE falls within the interval centered at the steady-state MSE, defined as the time average MSE from the 500<sup>th</sup> up to the 1000<sup>th</sup> iteration, and of size two times the standard deviation of these steady-state MSE. Then, the convergence iteration for various values of  $\bar{\gamma}_2$  is described in Table 4.2. Regarding the update rates and NUE results, they are shown also in Table 4.2 and Figure 4.4, respectively.

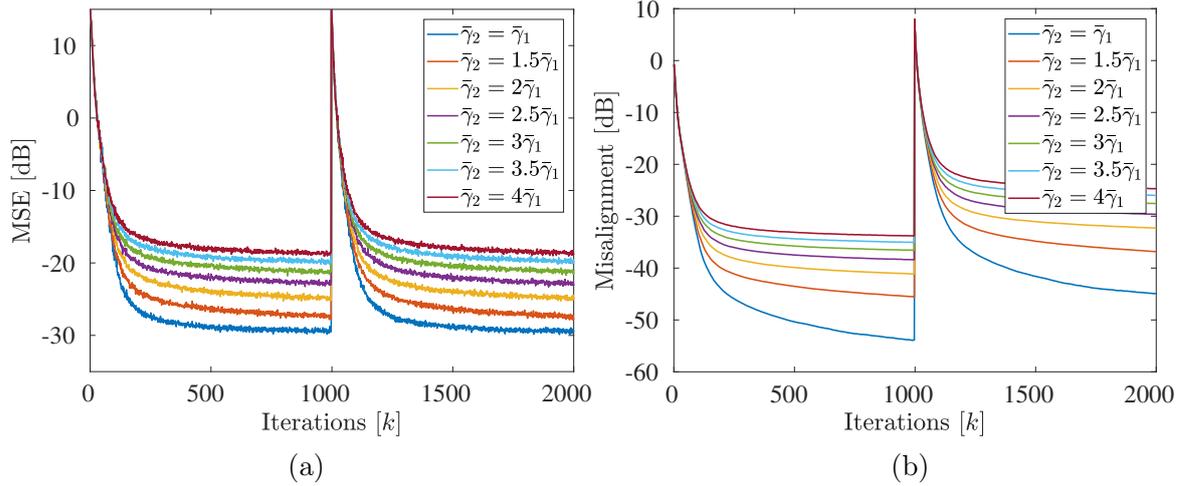


Figure 4.3: MSE (a) and misalignment (b) for VDTSM-PNLMS algorithm.

Table 4.2: Iterations until convergence and update rates for VDTSM-PNLMS algorithm using different thresholds  $\bar{\gamma}_2$ .

	Convergence iteration	$\nu_1$	$\nu_2$
$\bar{\gamma}_2 = \bar{\gamma}_1$	356	0%	14.02%
$\bar{\gamma}_2 = 1.5\bar{\gamma}_1$	419	11.81%	10.85%
$\bar{\gamma}_2 = 2\bar{\gamma}_1$	380	24.55%	9.79%
$\bar{\gamma}_2 = 2.5\bar{\gamma}_1$	367	35.18%	9.00%
$\bar{\gamma}_2 = 3\bar{\gamma}_1$	346	42.57%	8.47 %
$\bar{\gamma}_2 = 3.5\bar{\gamma}_1$	361	48.33%	7.98 %
$\bar{\gamma}_2 = 4\bar{\gamma}_1$	365	52.52%	7.61 %

## Conclusions of System Identification Simulations

The results presented in Figure 4.3 show that the steady-state MSE and misalignment levels increase as  $\bar{\gamma}_2$  grows. As expected, the update rate  $\nu_2$  decreases when  $\gamma_2$  increases, but also leads to an increasing in the update rate  $\nu_1$ . In addition, by analyzing Figure 4.4, where the red horizontal line indicates  $\text{NUE}_S$ , one is able say that VDTSM-PNLMS was less efficient than VSM-PNLMS. This is due to the fact that for all values of  $\bar{\gamma}_2$  employed, none of them satisfied (4.32). Nonetheless, the result for  $\bar{\gamma}_1 = \bar{\gamma}_2$  present similar steady-state MSE and misalignment if compared to VSM-PNLMS's, reducing the computational burden of Volterra series. The forthcoming results assess the performance of the proposed Volterra techniques in an equalization scenario.

### 4.4.4 Equalization Results for VDTSM-PNLMS Algorithm

Regarding feedforward equalization. Figure 4.5 and 4.6 show the MSE and BER results for  $M_{\text{FF}} \in \{3, 4\}$ , and employing different values of  $\bar{\gamma}_2$ . Considering only Channel 1, the convergence speed was evaluated using the same procedure of Subsec-

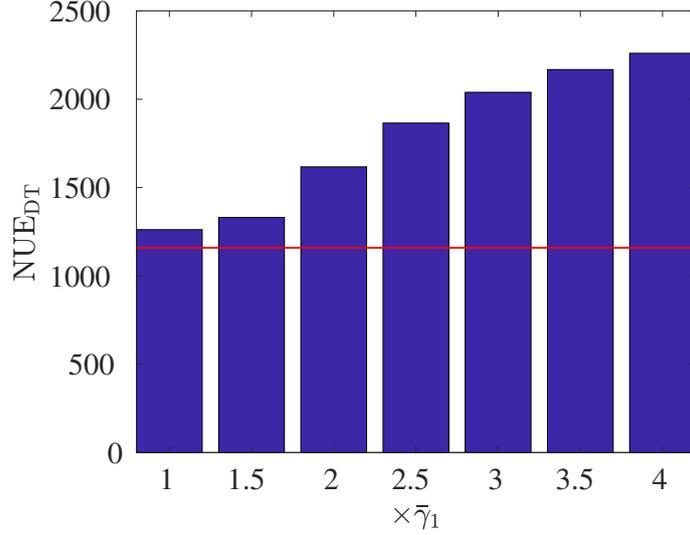


Figure 4.4: NUE for VDTSM-PNLMS algorithm using different values of  $\bar{\gamma}_2$ . The red line indicates the number of updated elements in the VSM-PNLMS case.

tion 3.4.4, whose results are illustrated by Table 4.3 for different thresholds  $\bar{\gamma}_2$ . Regarding the update rates  $\nu_1$  and  $\nu_2$ , the results using VDTSM-PNLMS algorithm are also displayed in Table 4.3, and NUE results are displayed in Figure 4.7, where the blue and red horizontal lines represent the VSM-PNLMS's NUE for  $M_{\text{FF}} \in \{3, 4\}$ , respectively.

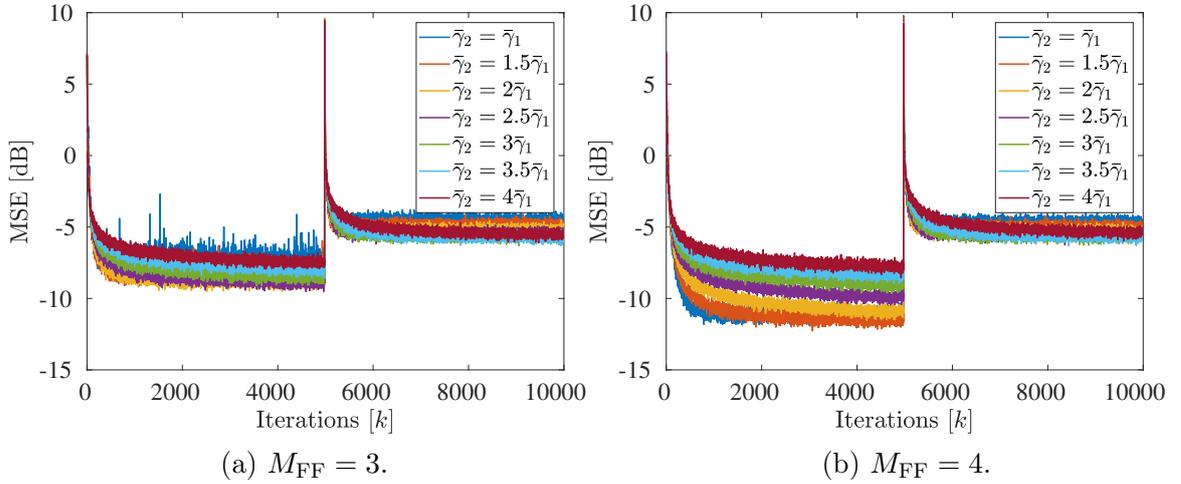


Figure 4.5: MSE for different memory sizes in the feedforward equalization case for VDTSM-PNLMS algorithm.

Considering a decision-feedback equalization scenario, where Volterra series is employed only in the feedforward filter, Figure 4.8 shows the MSE and BER results for VDTSM-PNLMS algorithm using  $M_{\text{FF}} = 4$  and  $M_{\text{FB}} = 0$ , as in the VSM-PNLMS case. Table 4.4 exhibits the convergence iteration and update rates, whereas Figure 4.9 shows the total updated elements in a single run of the VDTSM-PNLMS technique.

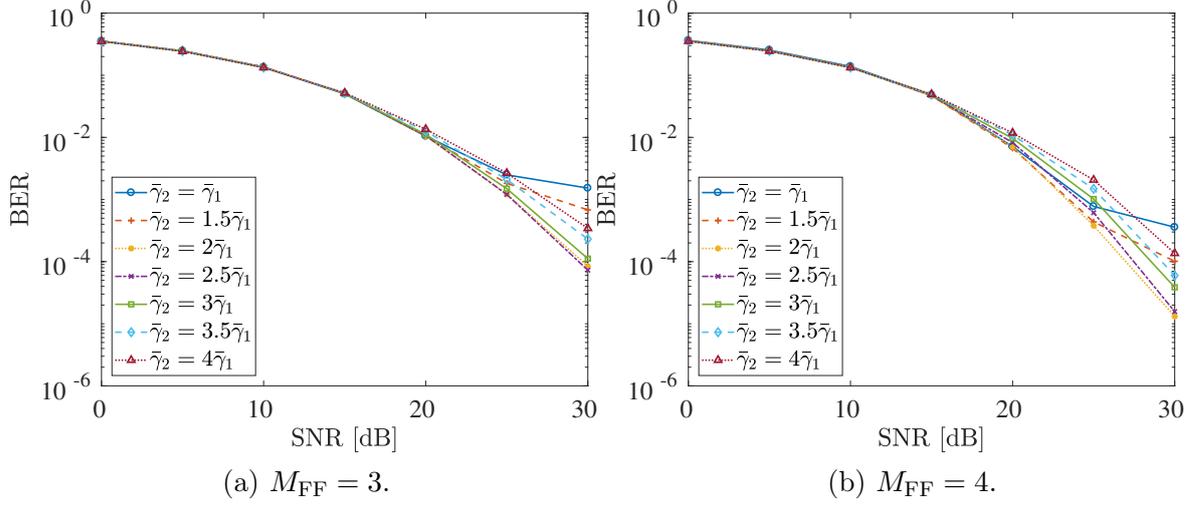


Figure 4.6: BER for different memory sizes in the feedforward equalization case for VDTSM-PNLMS algorithm.

Table 4.3: Iterations until convergence and update rates for  $M_{FF} = 3/M_{FF} = 4$ .

	Convergence iteration	$\nu_1$	$\nu_2$
$\bar{\gamma}_2 = \bar{\gamma}_1$	145 / 432	0 / 0%	43.95 / 28.55%
$\bar{\gamma}_2 = 1.5\bar{\gamma}_1$	267 / 846	18.57 / 16.47%	24.01 / 13.92%
$\bar{\gamma}_2 = 2\bar{\gamma}_1$	391 / 1490	30.83 / 28.17%	12.55 / 8.04%
$\bar{\gamma}_2 = 2.5\bar{\gamma}_1$	731 / 1419	38.79 / 37.06%	6.70 / 5.04%
$\bar{\gamma}_2 = 3\bar{\gamma}_1$	1043 / 1256	44.36 / 42.87%	3.83 / 3.24%
$\bar{\gamma}_2 = 3.5\bar{\gamma}_1$	868 / 1029	48.55 / 46.56%	2.35 / 2.15%
$\bar{\gamma}_2 = 4\bar{\gamma}_1$	913 / 1253	51.32 / 49.08%	1.53 / 1.49%

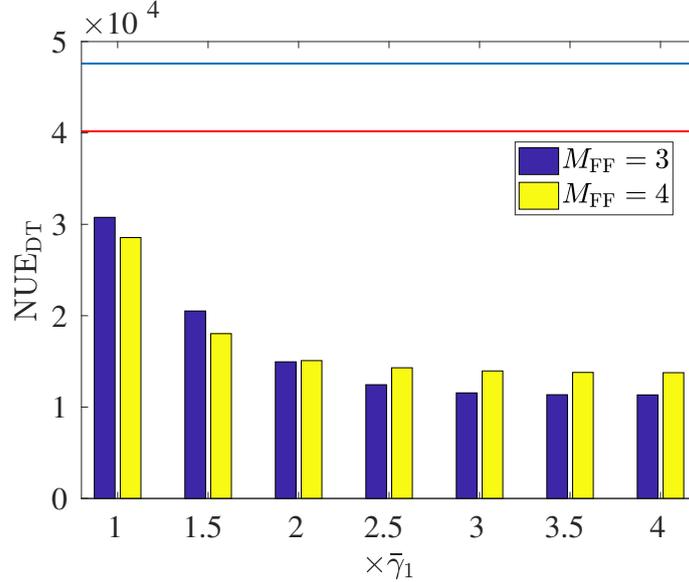


Figure 4.7: NUE for VDTSM-PNLMS algorithm in the feedforward equalization case using different values of  $\bar{\gamma}_2$ . The blue and red lines represent the NUE in VSM-PNLMS case for  $M_{FF} = 3$  and  $M_{FF} = 3$ , respectively.

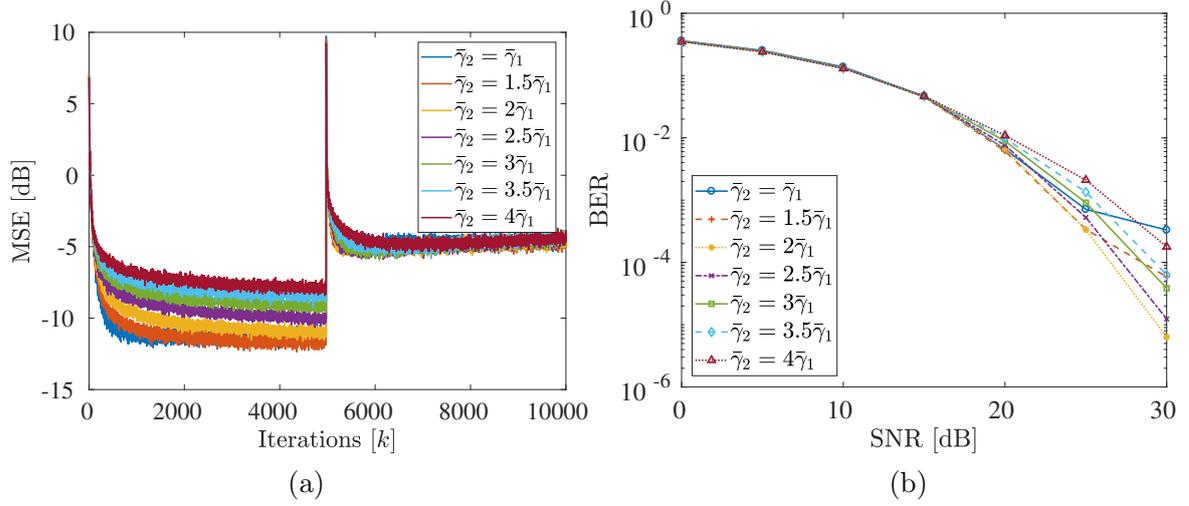


Figure 4.8: MSE (a) and BER (b) for  $M_{\text{FF}} = 4$  and  $M_{\text{FB}} = 0$  using different thresholds  $\bar{\gamma}_2$  in the DFE case for VDTSM-PNLMS.

Table 4.4: Iterations until convergence and update rates for  $M_{\text{FF}} = 4$  and  $M_{\text{FB}} = 0$ .

	Convergence iteration	$\nu_1$	$\nu_2$
$\bar{\gamma}_2 = \bar{\gamma}_1$	378	0%	27.96%
$\bar{\gamma}_2 = 1.5\bar{\gamma}_1$	974	16.49%	13.47%
$\bar{\gamma}_2 = 2\bar{\gamma}_1$	1355	28.27%	7.69%
$\bar{\gamma}_2 = 2.5\bar{\gamma}_1$	1531	37.10%	4.78%
$\bar{\gamma}_2 = 3\bar{\gamma}_1$	1217	42.62%	3.03%
$\bar{\gamma}_2 = 3.5\bar{\gamma}_1$	1296	46.34%	2.03%
$\bar{\gamma}_2 = 4\bar{\gamma}_1$	1178	48.72%	1.41%

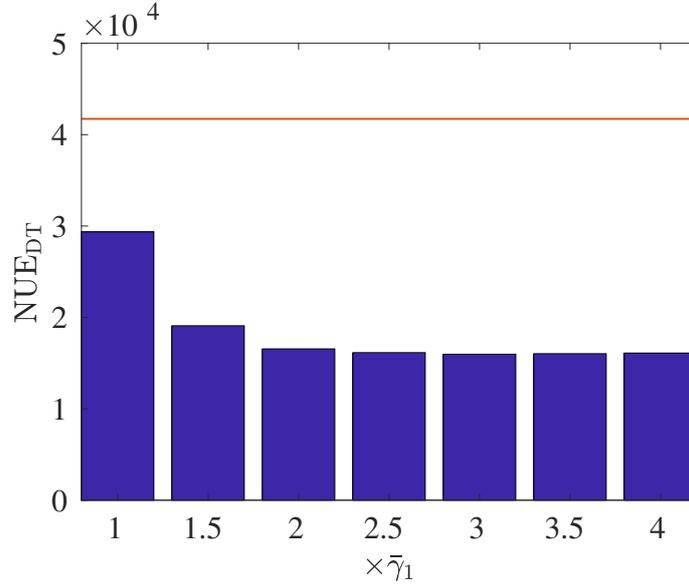


Figure 4.9: NUE for VDTSM-PNLMS algorithm in the DFE equalization case using different values of  $\bar{\gamma}_2$ . The red line represents the NUE in the VSM-PNLMS case.

### Conclusions of VDTSM-PNLMS Channel Equalization Simulations

Regarding feedforward equalization results, Figures 4.6(a) and 4.6(b) indicate that for all values of  $\bar{\gamma}_2$  employed, the resulting BERs are satisfactory for an SNR of

30 dB, specially using  $\bar{\gamma}_2 = \{2, 2.5\} \times \bar{\gamma}_1$ , which lead to the lowest BERs in both Figures 4.6(a) and 4.6(b). Indeed, by using  $\bar{\gamma}_2 = 2\bar{\gamma}_1$ , a BER of  $1.36 \times 10^{-4}$  was achieved for SNR = 30 dB and  $M_{FF} = 4$ , which is better than the result of VSM-PNLMS, and with a number of updated elements of 37.54% of VSM-PNLMS's, which is an enormous reduction in the computational complexity, yet enhancing the results in terms of BER. In addition, the speed of convergence of the double-threshold technique is almost 40% faster than VSM-PNLMS (1355 of convergence iteration against 2140, which indicates that this method outperforms the results of VSM-PNLMS considering the four figures of merit assessed in this chapter: MSE, BER, speed of convergence and average number of updated elements.

DFE equalization results show, once again, that all values of  $\bar{\gamma}_2$  yield satisfactory BER results, with  $\bar{\gamma}_2 = \{2, 2.5\} \times \bar{\gamma}_1$  leading to the lowest BER, as depicted in Figure 4.8(b). One should note that, this result once again outperforms VSM-PNLMS, besides being quite similar to VM-BEACON's, despite its quadratic complexity per update. Once more, considering  $\bar{\gamma}_2 = 2\bar{\gamma}_1$ , the NUE of VDTSM-PNLMS was only a fraction of VSM-PNLMS's, reducing drastically the computational burden associated with Volterra series, while improving the results in terms of BER.

#### 4.4.5 Results for VSBSM-PNLMS and VSBM-BEACON Algorithms

The MSE and BER results of feedforward equalization using these techniques are illustrated by Figures 4.10(a) and (b), in which the performance of the algorithms VSBSM-PNLMS and VSBM-BEACON are compared. Regarding the update rates and *Blind iteration*, the results are shown in Table 4.5. Now considering a DFE equalization scheme, Figure 4.10(a) displays MSE results, while Figure 4.10(b) shows BER results. Table 4.6 depicts the *Blind iteration* results and update rates for both VSBSM-PNLMS and VSBM-BEACON techniques.

Table 4.5: Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the feedforward case.

	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
<i>Blind iteration</i>	3367 / 459	1172 / 313	754 / 273
$\nu_{SP}$	63.59 / 63.56%	79.66 / 78.01%	84.33 / 82.69%
$\nu_{BP}$	40.19 / 30.60%	68.80 / 37.43%	71.17 / 49.98%
Overall update rate	51.32 / 32.48%	70.53 / 38.89%	72.47 / 50.97%

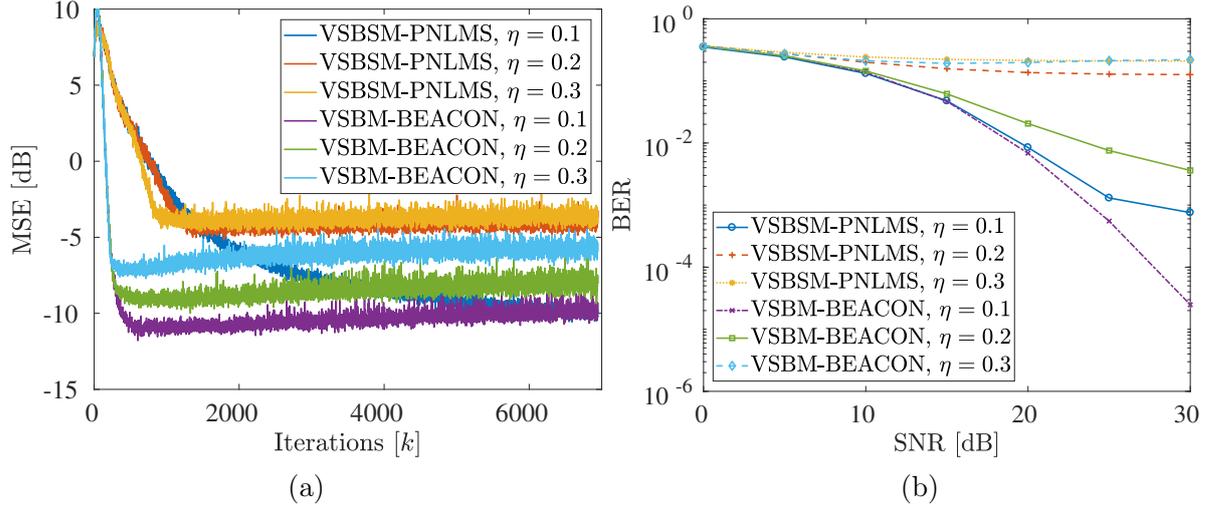


Figure 4.10: MSE (a) and BER (b) for  $M_{\text{FF}} = 8$  using different thresholds  $\bar{\gamma}_2$  in the feedforward case for VSBSM-PNLMS and VSBM-BEACON.

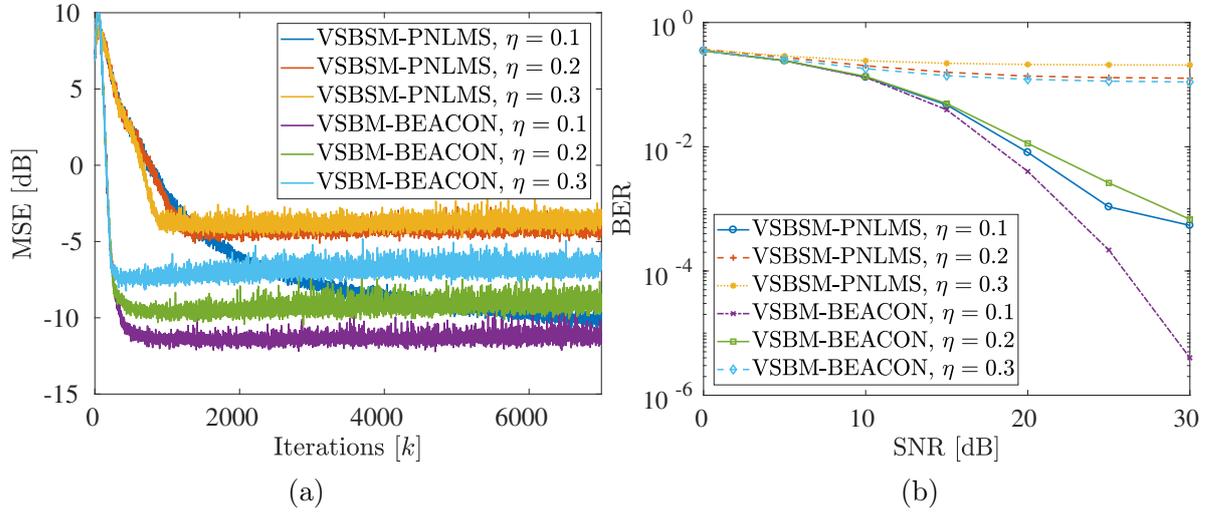


Figure 4.11: MSE (a) and BER (b) for  $M_{\text{FF}} = 8$  using different thresholds  $\bar{\gamma}_2$  in the DFE case for VSBSM-PNLMS and VSBM-BEACON.

Table 4.6: Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the DFE case.

	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
<i>Blind iteration</i>	3339 / 447	1159 / 310	760 / 272
$\nu_{\text{SP}}$	63.47 / 63.74%	79.76 / 77.94%	84.27 / 82.58%
$\nu_{\text{BP}}$	43.46 / 26.54%	68.67 / 33.16%	70.88 / 44.41%
Overall update rate	56.74 / 29.43%	71.13 / 35.39%	72.77 / 46.02%

### Conclusions of VSBSM-PNLMS and VSBM-BEACON Simulations

Considering the feedforward approach, it is possible to assert that the results of VSBM-BEACON reaches the lowest steady-state MSE level and BER, outperforming VSBSM-PNLMS's results. As mentioned before, one should take into consider-

ation that the complexity of VSBM-BEACON is quadratic in  $L$ , while it is linear in the VSBSM-PNLMS case. Considering  $\eta = 0.1$ , VSBSM-PNLMS employed, in average, 3339 symbols during training, while VSBM-BEACON used only 447 symbols, saving nearly 95% of the amount of training data, which represent huge bandwidth savings. For  $\eta \in \{0.2, 0.3\}$ , both algorithms do not present good BER results, possibly due to early switch from supervised period to unsupervised period. Nonetheless, considering the results for the Volterra semi-blind M-BEACON for  $\eta = 0.1$  and SNR = 30 dB, it was reached a level of BER quite similar to VM-BEACON case, which is, for sure, encouraging results, because a reduction of more than 80% of the amount of training data was achieved, with a technical draw regarding the update rates (29.43% vs. 30.58%).

By analyzing DFE results, the same conclusions of the feedforward case can be applied to this one, which indicates that in both equalization branches there were improvements when using the data-selective semi-blind approach. Nonetheless, it is worth pointing out that for  $\eta = 0.1$  and using VSBM-BEACON, the BER for an SNR = 30 dB is slightly lower than in feedforward case.

#### 4.4.6 Results for VSBDTSM-PNLMS Algorithms

The following results explore the impact of the thresholds  $\bar{\gamma}_2$  and  $\eta$  on both feedforward and DFE equalization cases when using the Volterra semi-blind double-threshold SM-PNLMS algorithm. Figure 4.12 shows the average MSE from the 6000<sup>th</sup> up to the 7000<sup>th</sup> iteration for diverse values of  $\eta$  and  $\bar{\gamma}_2$ , and using feedforward 4.12(a) and DFE 4.12(b) equalizations. As indicated by these results,  $\eta = 0.1$  leads to lower average MSE levels, hence, the following results consider this value.

Figures 4.13 and 4.15 show the MSE and BER results for the same previous values of  $\bar{\gamma}_2$ , and for feedforward and DFE equalization schemes, whereas Tables 4.7 and 4.8 describe the *Blind iteration* and update rates for the aforementioned equalization branches. These tables also show the update rates during the supervised and unsupervised periods for both filters  $\mathbf{w}_n[k]$ ,  $i \in \{1, 2\}$ , and denoted as  $\nu_{SP_n}$  and  $\nu_{BP_n}$ , correspondingly. Regarding the NUE of both equalization scenarios, they are displayed in Figures 4.14 and 4.16.

#### 4.4.7 Conclusions of VSBDTSM-PNLMS Simulation

Regarding feedforward equalization, the values of threshold  $\bar{\gamma}_2$  which yield the lowest BER are  $\bar{\gamma}_2 = \{1.5, 2\} \times \bar{\gamma}_1$ . Comparing Figures 4.13(b) and 4.10(b) the semi-blind double-threshold technique yields similar BER results, even comparing to the VSBSM-BEACON's. Regarding the performance of VSBSM-PNLMS, the results here are not even comparable, due to the huge reduction in the BER levels. With

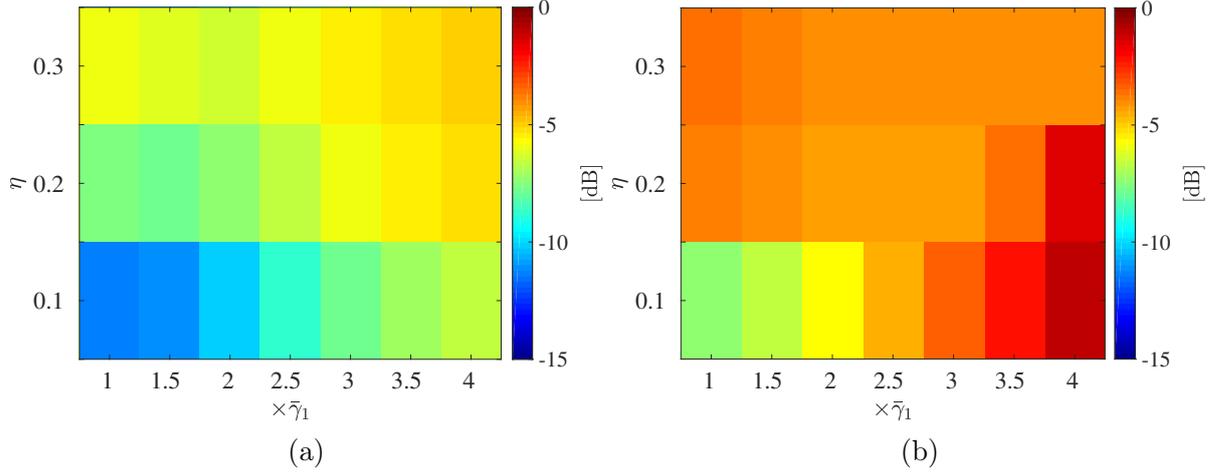


Figure 4.12: Steady-state MSE for different values of  $\bar{\gamma}_2$  and  $\eta = 0.1$  in the feedforward (a) and DFE (b) cases for VSB DTSM-PNLMS.

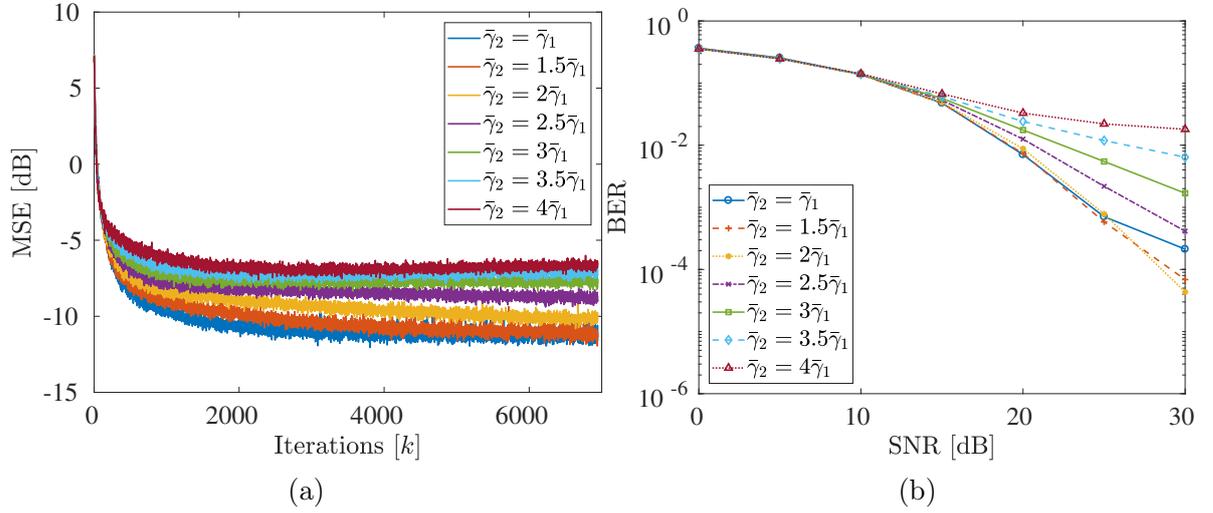


Figure 4.13: MSE (a) and BER (b) for  $M_{FF} = 8$ ,  $\eta = 0.1$ , and using different thresholds  $\bar{\gamma}_2$  in the feedforward case for VSB DTSM-PNLMS.

Table 4.7: Average number of iterations until unsupervised period and update rates using  $\eta = 0.1$  for VSB DTSM-PNLMS in the feedforward case.

	<i>Blind iteration</i>	$\nu_{SP_1}$	$\nu_{BP_1}$	$\nu_{SP_2}$	$\nu_{BP_2}$	$\nu_1$	$\nu_2$
$\gamma_2 = \gamma_1$	685	0%	0%	53.44%	28.41%	0%	30.65%
$\gamma_2 = 1.5\gamma_1$	861	17.22%	17.24%	35.59%	15.13%	17.24%	17.48%
$\gamma_2 = 2\gamma_1$	1166	30.14%	29.67%	22.27%	9.24%	29.74%	11.31%
$\gamma_2 = 2.5\gamma_1$	1590	38.99%	38.82%	13.45%	6.46%	38.86%	8.00%
$\gamma_2 = 3\gamma_1$	1910	44.73%	45.40%	8.50%	4.10%	45.22%	5.27%
$\gamma_2 = 3.5\gamma_1$	2245	48.90%	51.00%	5.16%	0.79%	50.34%	2.17%
$\gamma_2 = 4\gamma_1$	2861	51.37%	53.23%	3.26%	0.36%	52.49%	1.53%

respect to the average number of symbols used for training, the VSB DTSM-PNLMS algorithm employed, for  $\bar{\gamma}_2 = 2\bar{\gamma}_1$ , just 1166 symbols, in average, decreasing in

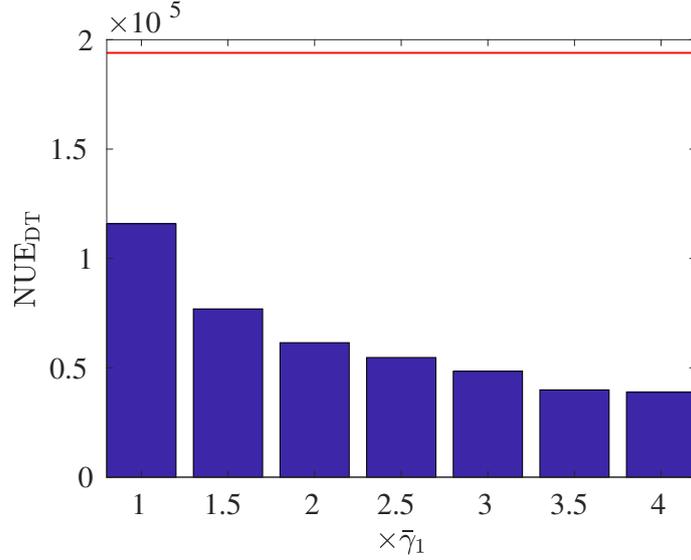


Figure 4.14: NUE for VSB DTSM-PNLMS algorithm in the feedforward equalization case using different values of  $\bar{\gamma}_2$ . The red line indicates the NUE in the VSBSM-PNLMS case.

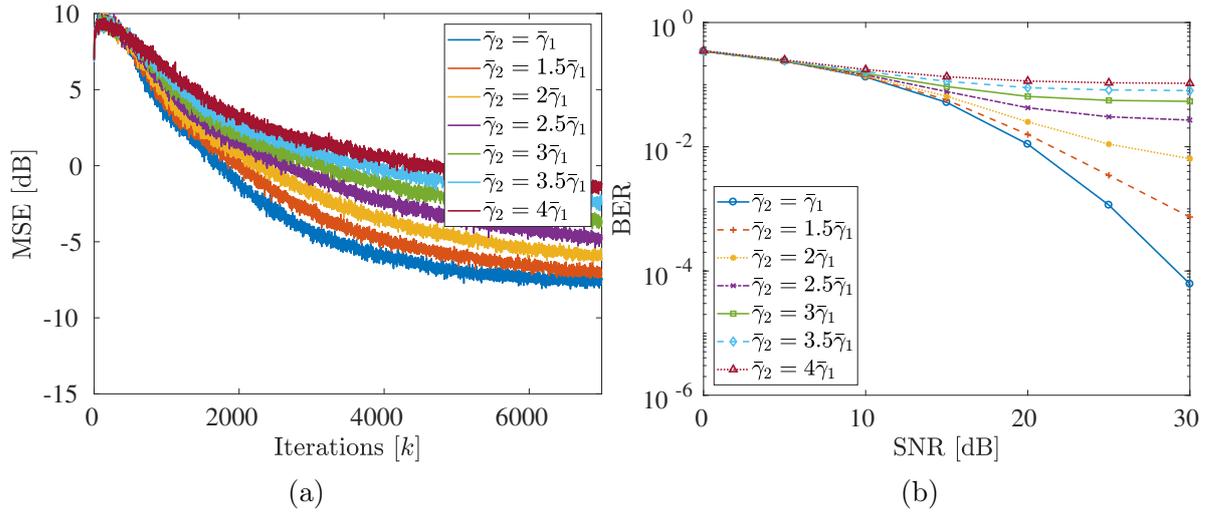


Figure 4.15: MSE (a) and BER (b) for  $M_{\text{FF}} = 8$ ,  $\eta = 0.1$ , and using different thresholds  $\bar{\gamma}_2$  in the DFE case for VSB DTSM-PNLMS.

approximately 85% the training data, indicating that the connection of the theory of combination of filters with data-selective semi-blind schemes yield promising results.

For all values of  $\bar{\gamma}_2$ , VSB DTSM-PNLMS's NUE is lower than in VSBSM-PNLMS case, as described in Figure 4.14. Moreover, for those two values of  $\bar{\gamma}_2$ , the resulting NUE represents only 39.66% and 31.70% of VSBSM-PNLMS', and yet leading to similar BER results. Considering the semi-blind features, VSB DTSM-PNLMS technique presents faster convergence if compared to VSBSM-PNLMS, besides using much less training symbols. Generally speaking, this algorithm yielded solid MSE and BER results, still reducing the computational complexity due to Volterra series and enhancing the spectral efficiency.

Table 4.8: Average number of iterations until unsupervised period and update rates using  $\eta = 0.1$  for VSB DTSM-PNLMS in the DFE case.

	<i>Blind iteration</i>	$\nu_{SP_1}$	$\nu_{BP_1}$	$\nu_{SP_2}$	$\nu_{BP_2}$	$\nu_1$	$\nu_2$
$\gamma_2 = \gamma_1$	5998	0%	0%	67.00%	50.35%	0%	64.46%
$\gamma_2 = 1.5\gamma_1$	6145	13.19%	18.97%	56.84%	36.84 %	13.95%	54.22%
$\gamma_2 = 2\gamma_1$	6311	22.92%	32.31%	49.85%	27.97%	23.93%	47.49%
$\gamma_2 = 2.5\gamma_1$	5888	29.16%	40.50%	47.15%	24.02%	31.07%	43.26%
$\gamma_2 = 3\gamma_1$	–	–	–	–	–	45.22%	40.31%
$\gamma_2 = 3.5\gamma_1$	–	–	–	–	–	50.34%	38.01%
$\gamma_2 = 4\gamma_1$	–	–	–	–	–	52.49%	36.10%

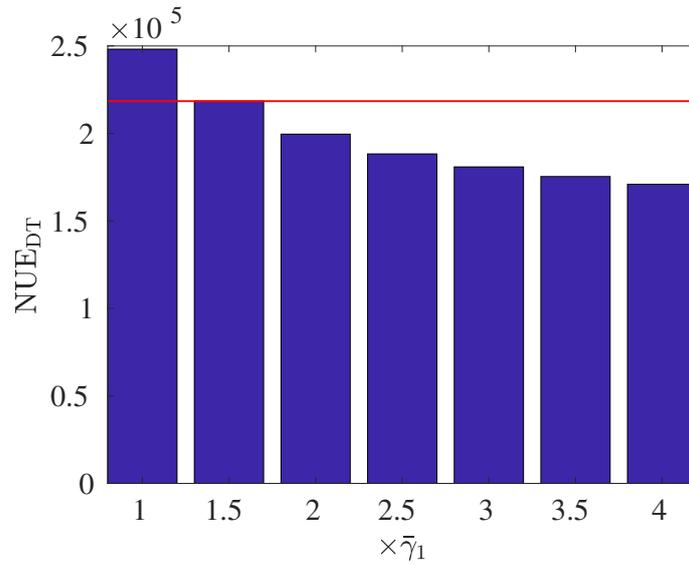


Figure 4.16: NUE for VSB DTSM-PNLMS algorithm in the DFE equalization case using different values of  $\bar{\gamma}_2$ . The red line indicates the NUE in the VSBSM-PNLMS case.

Unfortunately, in the DFE case, the VSB DTSM-PNLMS method did not present as good results as in the feedforward fashion, except for  $\bar{\gamma}_2 = \bar{\gamma}_1$ , which yields a BER similar to the lowest BER of feedforward case, but with a higher NUE, as indicated by Figure 4.16. In addition, as in VSBSM-PNLMS case, the adaptive filter converges slowly, but here with a steady-state MSE slightly higher. It is worth mentioning that for  $\bar{\gamma}_2 > 2.5\bar{\gamma}_1$ , the algorithm did not reach the unsupervised period, as indicated by dashes in Table 4.8, whose cause is the high steady-state MSE level. This should be further explored in other scenarios, where factors such as feedforward and feedback memories, or different nonlinear systems employed.

#### 4.4.8 Final Remarks

The proposed Volterra  $C$ -threshold and Volterra semi-blind algorithms represent good solutions to address the high computational complexity due to Volterra series. Furthermore, considering semi-blind techniques, they are also an interesting approach to enhance the spectral efficiency. Therefore, all data-selective techniques with Volterra series proposed in Chapter 3 and in this chapter will be employed in the VLC context, so as to provide a reliable and efficient approach to equalize a VLC transceiver, whose main components are modeled in the next chapter.

# Chapter 5

## Modeling the Key Components of a VLC System

This chapter describes a computational platform for visible light communication. The motivation for developing this simulator relies on the lack of such platform in the literature. Besides, such computational tool helps assess the equalization performance of the proposed filters of Chapters 3 and 4 within a VLC framework. Through this chapter, the key elements of the system described in Chapter 2 are modeled and properly interconnected so as to simulate a VLC system. This platform, for instance, can be used to analyze the bit error rate (BER) for an on-off keying (OOK) under a given bit rate and signal-to-noise ratio (SNR), or to compare the performance of two different modulation schemes, among other possible uses.

Let a baseband signal  $c[k]$  represent a pre-processed data signal mapped from a given modulation scheme (OOK, PAM, PPM, etc). Before passing through the LED,  $c[k]$  must be converted into either current or voltage, depending on the LED driving circuit. This work considers a voltage-based driving circuit, so that the LED input signal  $V_{\text{in}}[k]$  ( $V_{\text{in}}[k]$  represents the time-discrete version of  $V_{\text{in}}(t)$  introduced in Section 2.1) is described as

$$V_{\text{in}}[k] = V_{\text{DC}} + \alpha c[k], \quad (5.1)$$

where  $V_{\text{DC}} \in \mathbb{R}_+$  is a DC voltage bias added to ensure  $V_{\text{in}}[k]$  is nonnegative, and  $\alpha \in \mathbb{R}_+$  is a voltage constant, which is chosen to keep the LED input signal inside a predefined operational range. From (5.1), one can see that  $V_{\text{in}}[k]$  is a signal comprised of a DC part,  $V_{\text{DC}}$ , and an AC part,  $\alpha c[k]$ , where  $V_{\text{max}}$  will denote the maximum amplitude of  $V_{\text{in}}[k]$ . The so-called modulation index (MI) [43] quantifies

how much the amplitude of the data signal  $V_{\text{in}}[k]$  varies around  $V_{\text{DC}}$ , i.e.

$$\text{MI} = \frac{\Delta V}{V_{\text{DC}}} = \frac{V_{\text{max}} - V_{\text{DC}}}{V_{\text{DC}}}. \quad (5.2)$$

If MI and  $V_{\text{DC}}$  values are predefined parameters, then by using (5.1) and (5.2) one can write an expression for  $\alpha$  as

$$\alpha = \frac{\text{MI} \cdot V_{\text{max}}}{(\text{MI} + 1) \max\{c[k]\}}. \quad (5.3)$$

Once the LED input signal model was presented, the computational process to convert  $V_{\text{in}}[k]$  into a corresponding current level is modeled by the  $I$ - $V$  LED curve, as will be better explained in Section 5.1. The product of  $V_{\text{in}}[k]$  and its corresponding current is then translated into a certain level of luminous intensity signal,  $\mathcal{I}[k]$ , according to the LED luminous efficacy. After that,  $\mathcal{I}[k]$  is convolved with a linear time-invariant filter  $h[k]$ , which emulates the light signal propagation through the air (FSO channel), and then is converted into a corresponding current level  $r[k]$ . This current is corrupted by a zero-mean white Gaussian noise  $g[k]$  [88], and preamplified by the transimpedance amplifier (TIA). The resulting received signal is described by

$$r[k] = \beta \times (R\{A \cdot (\mathcal{I} * h)[k]\} + g[k]), \quad (5.4)$$

where  $*$  represents a linear convolution operation,  $A \in \mathbb{R}_+$  is the photodiode (PD) detection area,  $R\{\cdot\}$  denotes the PD responsivity [A/W], and  $\beta \in \mathbb{R}_+$  stands for the TIA gain. A simplified block diagram of a VLC system is shown in Figure 5.1. It is worth highlighting that the driving circuit is treated in this work as an ideal mapper of data into electrical signal, and does not impose any other undesired effect. Next section describes some computational models for LED, free-space optical channel, and receiver that will be incorporated in the computational platform.

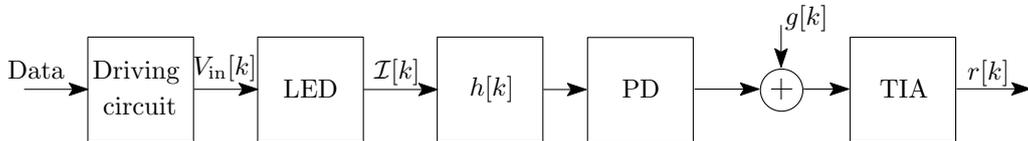


Figure 5.1: Simplified VLC model.

## 5.1 LED Model

The LED is usually the component that generates the most severe distortion effects on a VLC system. For instance, the LED may experience very limited bandwidth,

depending on its type. Moreover, the nonlinear relation between the voltage applied to the LED and the corresponding output current, as well as the nonlinear conversion of instantaneous electrical power to instantaneous luminous intensity are issues that will be modeled in this work. To do that, a low cost white pc-LED, whose datasheet can be found in [104], is considered.

### 5.1.1 Frequency-Response Model

Devices like pc-LEDs feature low-pass frequency response due to slow response of their phosphor layers, where the 3-dB bandwidth is usually around 2 MHz [81, 86, 105]. The frequency-response model adopted in this work is the one proposed in [105], where the authors performed an experiment to measure the frequency response  $F(\omega)$  of a commercial white LED, approximated as:

$$F(\omega) = \begin{cases} e^{-|\omega|/\omega_1}, & \text{if } |\omega| < \omega_c \\ e^{-|\omega_c|/\omega_1} e^{|\omega_c|/\omega_2} e^{-|\omega|/\omega_2}, & \text{otherwise,} \end{cases} \quad (5.5)$$

where  $\omega_1 = 2\pi \cdot 3.26$ ,  $\omega_2 = 2\pi \cdot 10.86$ , and  $\omega_c = 2\pi$  Mrad/s. In order to employ this frequency response model in a digital environment,  $F(\omega)$  should be truncated and sampled at a rate  $\omega_s \in \mathbb{R}_+$ , generating the discrete-time model  $F_s(e^{j\omega})$ .

It is worth mentioning that, even though the LED used in [105] is different from the one employed in this work, the frequency response modeled by (5.5) will be used here; nonetheless, the proposed computational platform encapsulates this part so that the user can easily change it.

### 5.1.2 $I$ - $V$ Curve Model

After defining the LED input signal and its frequency response models, it is necessary to model the behavior of the LED current as a function of the input voltage. There are some approximations for the current-voltage  $I$ - $V$  curve of LEDs, such as ideal, piecewise-linear, and exponential approximations [78]. This device can work in two modes of operation, as shown in Figure 5.2, viz.:

- Forward biased: current flows from the anode to the cathode whose magnitude is related to the positive voltage employed.
- Reverse biased: if a negative voltage is applied to the LED, no current flows from the anode to its cathode, and the device works approximately as open circuit.

In the ideal approximation, the LED has zero current when operating in reverse-bias mode (or reversely polarized). On the other hand, when current flows from

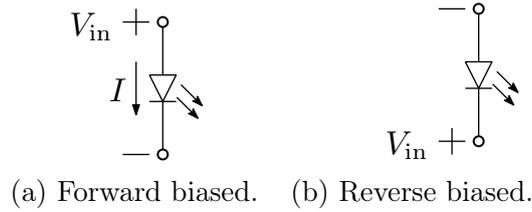


Figure 5.2: LED's modes of operation.

anode to cathode, the LED is said to be directly polarized, and there is no drop in voltage. The resulting  $I$ - $V$  curve is illustrated in Figure 5.3, as well as the equivalent model.

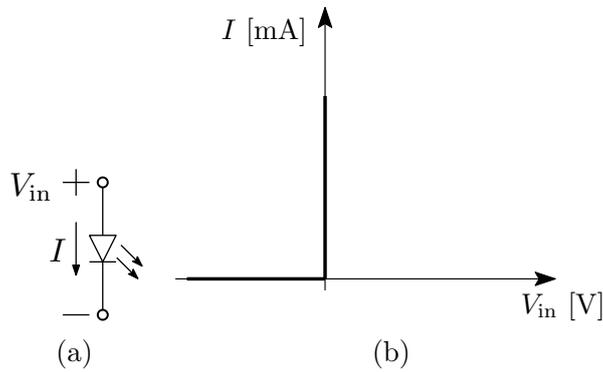


Figure 5.3: Equivalent model (a) and (b)  $I$ - $V$  curve for ideal approximation.

The second approximation is the piecewise linear, whose equivalent model is an ideal LED in series with an ideal voltage source  $V_D$  and a resistance  $R_D$ , as depicted in Figure 5.4(a). Therefore, the LED is forward biased if  $V_{in} > V_D$ , culminating in the  $I$ - $V$  curve of Figure 5.4(b), whose slope is given by  $\frac{1}{R_D}$ .

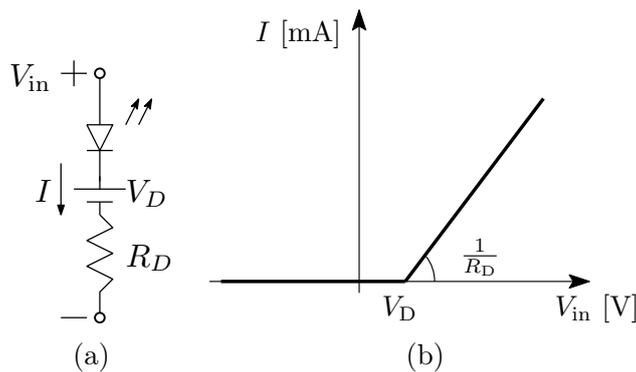


Figure 5.4: (a) Equivalent model and (b)  $I$ - $V$  curve for piecewise approximation.

The last model is the exponential approximation, where  $I(t)$  is an exponential function of  $V_{in}(t)$ , described mathematically as

$$I(t) = I_s \left( e^{\frac{V_{in}(t)}{n_{LED} V_T}} - 1 \right), \quad (5.6)$$

where  $I_s$  denotes the saturation current, which is proportional to the cross-sectional area of the LED and is usually in the order of micro amperes,  $n_{\text{LED}}$  is the ideality factor of the LED, which may vary from 2 up to 7 and is related to its fabrication process, and  $V_T$  represents the thermal voltage, which is expressed as

$$V_T = \frac{K_B T}{q}, \quad (5.7)$$

where  $K_B = 1.38 \times 10^{-23}$  joule/kelvin is the Boltzmann's constant,  $T \in \mathbb{R}_+$  is the absolute temperature in kelvin, and  $q = 1.6 \times 10^{-19}$  coulomb stands for the electron's charge magnitude. Thus, the  $I$ - $V$  curve for this approximation is illustrated in Figure 5.5.

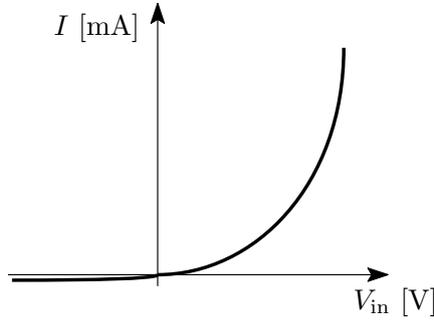


Figure 5.5:  $I$ - $V$  curve for exponential approximation.

Among the previous three approximations for the  $I$ - $V$  curve, the most realistic is the exponential model (compare it with Figure 2.2), once the other two models are better suited for digital or high amplitude signals. Therefore, one can modify (5.6) to work with positive voltages, and describe its discrete-time version as

$$I[k] = \begin{cases} I_s(e^{V_{\text{in}}[k]/n_{\text{LED}}V_T} - 1), & \text{if } V_{\text{in}}[k] \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

Note that the LED current grows exponentially with voltage; however, there is a region where current varies approximately linearly with voltage, thus suggesting that the operational point ( $V_{\text{DC}}$ ) should be set within this region. Then, the resulting instantaneous electrical power is

$$P_{\text{electrical}}[k] = V_{\text{in}}[k] \cdot I[k], \quad (5.9)$$

which now must be converted into optical power. As will be shown, this conversion may be nonlinear, depending on the level of the instantaneous electrical power.

### 5.1.3 Electrical-to-Optical Conversion Model

In [106] the authors proposed a model for the electrical-to-optical conversion of an LED, which is described as

$$\mathcal{I}[k] = \frac{P_{\text{electrical}}[k] \times \rho}{\left(1 + \left(\frac{P_{\text{electrical}}[k] \times \rho}{\mathcal{I}_{\text{max}}[k]}\right)^{2k}\right)^{1/2k}}, \quad (5.10)$$

where  $k \in \mathbb{R}_+$  is a knee factor that adjusts the smoothness of the luminous intensity saturation, and  $\mathcal{I}_{\text{max}} \in \mathbb{R}_+$  is the maximum luminous intensity emitted by the LED. In fact, when operating in the linear region of the  $I$ - $V$  curve, the luminous intensity of an LED is proportional to its output electrical power, i.e.,  $\mathcal{I}[k] = P_{\text{electrical}}[k] \times \rho$ ,  $\rho \in \mathbb{R}_+$ . Nevertheless, there is a certain level of electrical power where the luminous intensity saturates, i.e., part of the electrical power is dissipated as heat. Figure 5.6 illustrates the aforementioned effect for different values of  $k$  modeled by (5.10).

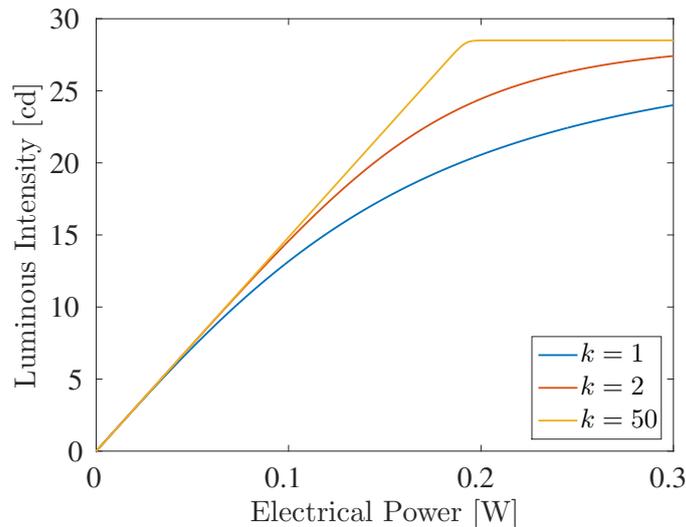


Figure 5.6: Nonlinear behavior of luminous intensity for different  $k$ 's.

Note that the nonlinear effect imposed by the electrical-to-optical conversion is directly connected to MI in (5.2). Indeed, when MI increases, the maximum allowed value for the LED input signal also increases, leading to a higher degree of nonlinearity imposed by the  $I$ - $V$  curve, and possibly, causing saturation in the LED's electrical-to-optical conversion.

Figure 5.7 outlines the complete model of the LED, assumed as a Wiener model, in which  $c[k]$  represents a symbol of a given constellation, and  $O\{\cdot\}$  denotes a nonlinear function that considers LED's  $I$ - $V$  relation as well as its electrical-to-optical conversion. When a nonlinear function is preceded by a linear filter, the resulting model is named Wiener model. On the other hand, when a filter is located after the nonlinear function, the resulting model is called Hammerstein model [107, 108].

These schemes are often used to model nonlinear systems, with applications in adaptive filtering [109], VLC systems [110], and nonlinear control systems [111, 112]. Regarding the chosen model, the first one will be used for it fits better LED's characteristics, since its frequency response depends on the modulation index level [43].

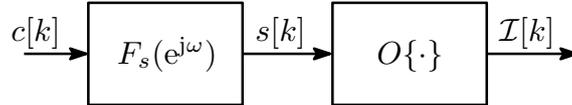


Figure 5.7: Complete LED model.

### 5.1.4 Optical Characteristics

With respect to the optical characteristics of LEDs, they can be regarded as a Lambertian transmitter, whose light propagation is well modeled by a Lambertian radiation pattern [83], illustrated in Figure 5.8 for distinct half-power angles, in which  $S$  represents a surface and  $\hat{\mathbf{n}}_S$  denotes its normal vector. The normalized radiant intensity of the LED,<sup>1</sup> which measures the amount of power per solid angle, is given by [17]:

$$R_o(\phi) = \left( \frac{m+1}{2\pi} \right) \cos^m(\phi), \quad (5.11)$$

where  $m = \frac{-\ln 2}{\ln \cos(\Phi_{1/2})}$ , with  $\Phi_{1/2} \in \mathbb{R}_+$  being the LED's half-power angle. One should notice that the emitted beam of light in this case is much more focused than the RF's (even though beamforming techniques may be used), once common LED's half-power angle are usually not greater than  $30^\circ$  [17]. It is worth pointing out that the radiation pattern is incorporated to the assessment of the channel gain, which will be further described.

After modeling the main effects imposed by an LED when it is working as a VLC transmitter, the next step is to consider the optical channel model and incorporate it to the computational platform developed in this work.

## 5.2 Free-Space Optical Channel Model

As discussed in Section 2.3, the FSO channel mostly depends on the channel DC gain  $H_0$ . Considering that the FSO channel features a line-of-sight (LOS), i.e., the direct path between transmitter and receiver is not obstructed, and the LED as a Lambertian transmitter, whose pattern is described in Figure 5.8, by using (5.11),

<sup>1</sup> $R_o(\phi)$  is normalized so that  $\int_0^\pi R_o(\phi) \sin(\phi) d\phi = 1$ .

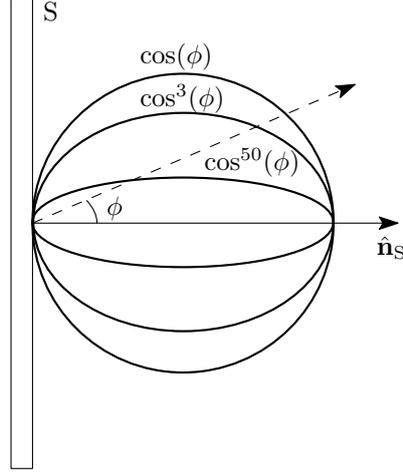


Figure 5.8: Shape of generalized Lambertian radiation pattern for different half-power angles.

one can describe  $H(0)$  as

$$H(0) = \begin{cases} \frac{AR_o(\phi)\cos(\theta)\mathcal{T}(\theta)\mathcal{G}(\theta)}{\zeta^2}, & \text{if } \left|\frac{\theta}{\text{FOV}}\right| \leq 1 \\ 0, & \text{otherwise,} \end{cases} \quad (5.12)$$

where  $\text{FOV} \in \mathbb{R}_+$  and  $A \in \mathbb{R}_+$  denote the photodiode (PD) field of view and area, respectively,  $\mathcal{T}(\theta)$  and  $\mathcal{G}(\theta)$  represent the optical filter and concentrator gains, and  $\zeta$  represents the distance between LED and PD.  $\mathcal{T}(\theta)$  may represent an average gain over the wavelengths and/or different incidence angles. As for  $\phi$  and  $\theta$ , consider Figure 5.9, in which the vectors  $\mathbf{r}_T, \mathbf{r}_R \in \mathbb{R}^{3 \times 1}$  respectively denote the LED and PD positions, and  $\hat{\mathbf{n}}_T, \hat{\mathbf{n}}_R \in \mathbb{R}^{3 \times 1}$  represent the normal vectors to their surfaces. Then,  $\phi$  is the angle between  $\hat{\mathbf{n}}_T$  and  $(\mathbf{r}_R - \mathbf{r}_T)$ , while  $\theta$  is the angle between  $\hat{\mathbf{n}}_R$  and  $(\mathbf{r}_R - \mathbf{r}_T)$ .

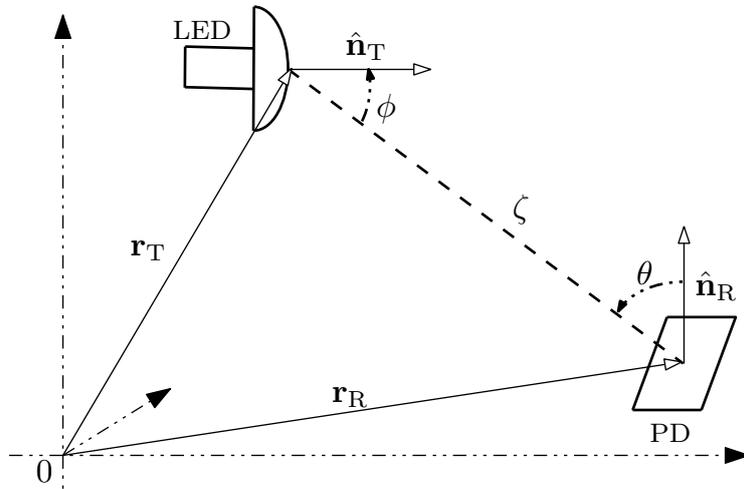


Figure 5.9: Geometry of the LED and PD.

Regarding non-line-of-sight optical (NLOS) channels,  $H(0)$  evaluation becomes cumbersome because all derivation is very sensitive to the geometry of the set transmitter-receiver, but more details can be found at [17]. With respect to IR channel simulators, one of the most popular is [113], whereas the authors in [114] propose a technique which considers the effects of the visible light absorption inherent to indoor finishing materials. Moreover, a suitable model for LOS channels considering multipath can be described as [17, 115]

$$h(t) = H(0) \frac{6a^6}{(t+a)^7} u(t), \quad (5.13)$$

where  $a = 12\sqrt{11/13} D_s$ , with  $u(t)$  denoting the unit step function, and  $D_s$  representing the root mean square (RMS) delay spread of the channel, given by

$$D_s = \sqrt{\left( \frac{\int_{-\infty}^{\infty} (t-\xi)^2 h^2(t) dt}{\int_{-\infty}^{\infty} h^2(t) dt} \right)}, \quad (5.14)$$

$$\xi = \frac{\int_{-\infty}^{\infty} t h^2(t) dt}{\int_{-\infty}^{\infty} h^2(t) dt}. \quad (5.15)$$

In practice, if one wants to generate a channel using (5.13), a delay spread  $D_s$  must be chosen in advance. The purpose of (5.14) is to evaluate  $D_s$  when the channel coefficients are known beforehand.

Considering a LOS channel, the value of the delay spread is often in the order of nanoseconds. Hence, the intersymbol interference (ISI) effects imposed by this channel occur more severely when the VLC system is working in a bit rate of few Gbps, which is not the case here, since it is considered bit rates in the order of Mbps. Therefore, the DC gain  $H(0)$  bears all the effects imposed by the optical channel. Consequently,

$$h[k] \approx \frac{2\pi}{\omega_s} H(0) \delta[k], \quad (5.16)$$

where  $\delta[k]$  denotes a discrete unit impulse. Then, considering that both optical filter and concentrator gain present unitary gain for all values of  $\theta$ , i.e.,  $\mathcal{T} = \mathcal{G} = 1$ , one

can describe the DC channel gain model employed in this work as

$$H(0) = \begin{cases} \frac{AR_o(\phi)\cos(\theta)}{\zeta^2}, & \text{if } \left|\frac{\theta}{\text{FOV}}\right| \leq 1 \\ 0, & \text{otherwise,} \end{cases} \quad (5.17)$$

It is important to highlight that, as in the RF case, the optical channel is also corrupted by an additive noise, which is modeled at the receiver as zero mean white Gaussian noise [88].

### 5.3 Receiver Model

One of the main characteristics of a photodiode is its response, i.e., associated gain, with respect to the wavelength of the incident light. This response, named responsivity, often refers to the peak sensitivity, i.e., to the wavelength which yields the maximum responsivity. Considering the photodiode in the photoconductive mode, the responsivity can be described as

$$R(\lambda, t) = \frac{I_p(\lambda, t)}{P_r(\lambda, t)}, \quad (5.18)$$

where  $I_p(\lambda, t)$  is the resulting current from the received power  $P_r(\lambda, t)$ .

The  $I$ - $V$  relation in the photodiode is similar to the one presented by (5.6) when there is no incident light. Supposing that a certain level of light illuminates the PD, generating a current  $I_p$ , (5.6) can be modified as follows

$$I(\lambda, t) = I_s \left( e^{\frac{V_{in}(t)}{nV_T}} - 1 \right) - I_p(\lambda, t). \quad (5.19)$$

Therefore,  $I_p(\lambda, t)$  only shifts vertically the  $I$ - $V$  curve, which is shown in Figure 5.10 for the photoconductive mode considering distinct levels of incident power, represented by  $P_0$ ,  $P_1$ , and  $P_2$ , with  $P_0 < P_1 < P_2$ . Once this work assumes the PD working in the photoconductive mode, the total current of this device will be considered as  $I_p$ , which is given by (5.18).

The receiver model employed in this work is represented only by the gain  $R$ , which was regarded as time-invariant and constant over the entire spectrum. This was assumed for the sake of simplicity and does not affect the generality of this work. Besides, the bandwidth of the PD was considered wider than the LED's. Regarding the model for the TIA positioned at the PD's output, it was considered that it works also as a gain, and its bandwidth was assumed to be larger than the LED's. Therefore, the dominant frequency selectivity of the VLC system modeled in this work is induced by the LED, whose low-pass frequency response model was

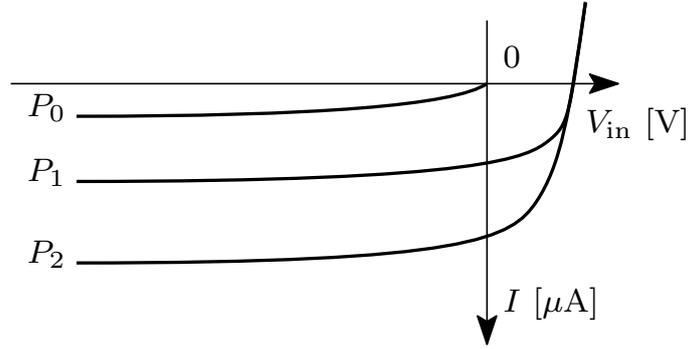


Figure 5.10:  $I$ - $V$  curve in reverse-bias mode for different values of incident power.

presented in Subsection 5.1.1. With respect to TIA's gain, it was set to match the variances of the received and transmitted signals, since their magnitudes are remarkably different, which can hinder the equalization processing.

Once the models for each key component were introduced, they can be properly interconnected to assemble a VLC transceiver. Nonetheless, this simulator should work as an analysis tool for the impact of diverse factors, e.g, modulation index level, LED frequency response, among others. To do that, the figure of merit employed in this work to inspect these factors is the bit error rate (BER) at the receiver, which reflects directly how the transceiver harms the data sent by the transmitter. Moreover, a good and reliable communication system is based on an adequate equalization scheme. As pointed out in Chapter 2, a VLC system experiences a considerable level of nonlinearity on the transmitted data. Hence, suitable equalizers must be employed so as to mitigate these effects. Therefore, in the next chapter, the proposed equalization techniques described in Chapters 3 and 4 will be applied in the task of equalizing the signals provided by the VLC simulator, which is the main focus of this work. Therefore, one will be able to assess the behavior of the complete VLC system.

# Chapter 6

## Results and Discussions

This chapter assesses the performance of the proposed data-selective techniques of Chapters 3 and 4 considering the equalization of the VLC transceiver, whose main elements were first introduced in Chapter 2, and digitally modeled in Chapter 5. Throughout this chapter, the methodology and figures of merit regarding the simulations are presented, as well as their setup, describing the parameters of the VLC simulator and MSE and BER simulations. At the end of this chapter, an outline concerning the results is presented.

### 6.1 Performance Evaluation

This section evaluates the performance of the proposed data-selective techniques described in Chapters 3 and 4. are described in Table 6.1. First, the performance of the supervised filters, i.e., the filters which are continually trained using a pilot sequence, is assessed. After that, the results of semi-blind algorithms are described. It is worth mentioning that the results must be properly analyzed according to the following perspective: there is a trade-off between computational effort, amount of pilot sequence, and performance, which depends on the application. For instance, if an adaptive filter is deployed in an embedded system, the computational complexity may be the main concern, while in a system where the bandwidth is scarce, it is interesting to allocate less data to train an adaptive filter.

#### 6.1.1 Simulation Methodology and Figures of Merit

The simulation methodology follows the ones presented in Subsection 3.4.1 and 4.4.1. In this chapter, the proposed methods are also evaluated under different values of modulation indexes (MIs). As indicated in Chapter 5, the modulation index controls the signal excursion over the  $I$ - $V$  curve of the LED, which means that a higher MI yields a higher nonlinearity level on the VLC transceiver. Thus, by varying MI allows

Table 6.1: Description of the proposed algorithms and sections in which they were presented.

	Description
VSM-PNLMS	Volterra set-membership PNLMS, 3.2.1
VM-BEACON	Volterra modified BEACON, 3.2.2
VSBSM-PNLMS	Volterra semi-blind set-membership PNLMS, 4.1.1
VSBM-BEACON	Volterra semi-blind modified BEACON, 4.1.2
VDTSM-PNLMS	Volterra double-threshold set-membership PNLMS, 4.2
VSBDTSM-PNLMS	Volterra semi-blind double-threshold set-membership PNLMS, 4.3

for the evaluation of the proposed techniques under distinct nonlinear scenarios. With respect to the figures of merit, they are identical to the ones described in Subsection 4.4.1.

## 6.1.2 Simulation Setup

The simulation setup of this chapter is organized as follows:

- VLC simulator's setup: The DC bias voltage  $V_{DC}$  in order to ensure the transmitted signals are nonnegative was 3.25 V. This value was also chosen so as to place the operational point around the middle of the linear part of the LED's  $I$ - $V$  curve. It was adopted a low cost pc-LED with half-power angle of  $\Phi_{1/2} = 15^\circ$  whose datasheet can be found at [104]. The distance between LED and photodiode was 10 cm, which were treated as perfectly aligned, leading to  $\phi = \theta = 0^\circ$ . Regarding the photodiode parameters, the responsivity, area, and FOV were set as  $R = 0.5$ ,  $A = 1$  cm, and  $FOV = 25^\circ$ , respectively. The knee-factor that controls the nonlinearity level in the electrical-to-optical conversion (see (5.10)) was  $k = 2$ , which induces a high degree of nonlinearity in this process.
- Equalization setup: during the training of the adaptive filters, it was used 1000 independent runs, a signal-to-noise ratio (SNR) of 30 dB, and 4-PAM symbols as input signals with a bandwidth of 2 MHz. Regarding proportionate algorithms,  $\kappa$  was set to 0.5, with  $\kappa_1 = \kappa_2 = \kappa$ . Considering the techniques presented in Chapter 3, the error threshold  $\bar{\gamma}$  was set to 0.28, while for the algorithms described in Chapter 4, the error thresholds were chosen as  $\bar{\gamma}_1 = \bar{\gamma}$  and with  $\bar{\gamma}_2$  varying from  $\bar{\gamma}_2 = \bar{\gamma}_1$  to  $\bar{\gamma}_2 = 4\bar{\gamma}_1$  using an interval of 0.5. In addition, the combination parameter  $\sigma$  was chosen as 0.5. With respect to semi-blind algorithms, the threshold employed to indicate the unsupervised period start was set as  $\eta \in \{0.1, 0.2, 0.3\}$ . The second-order Volterra series was used in the input of the adaptive filters, whereas their memory-sizes were

set to  $M = 8$ ,  $M_{\text{FF}} = 8$  and  $M_{\text{FB}} = 0$ , in the cases of feedforward and DFE equalization, respectively. Considering the DFE, as indicated by the results of Chapter 3, the Volterra series was applied only to the feedforward filter. In addition, the delay in samples imposed during the training of the adaptive filters was  $M + 2$  or ( $M_{\text{FF}}$  in the DFE case). For BER simulations, 10000 symbols were transmitted in each of the 1000 Monte Carlo loops.

### 6.1.3 Results of VSM-PNLMS and VM-BEACON Algorithms

Considering different values of MI, Figures 6.1(a) and 6.1(b) show the MSE and BER results, respectively, in the feedforward equalization scheme. Table 6.2 displays the convergence speed, which was evaluated using a procedure similar to the one presented in Subsection 3.4.3, but here using the time average MSE from the 4000<sup>th</sup> up to the 5000<sup>th</sup> iteration, and shows the update rates of the adaptive filters for both VSM-PNLMS and VM-BEACON techniques. Analogously, Figures 6.2(a) and 6.2(b), and Table 6.3 describe the results for a DFE equalization scenario.

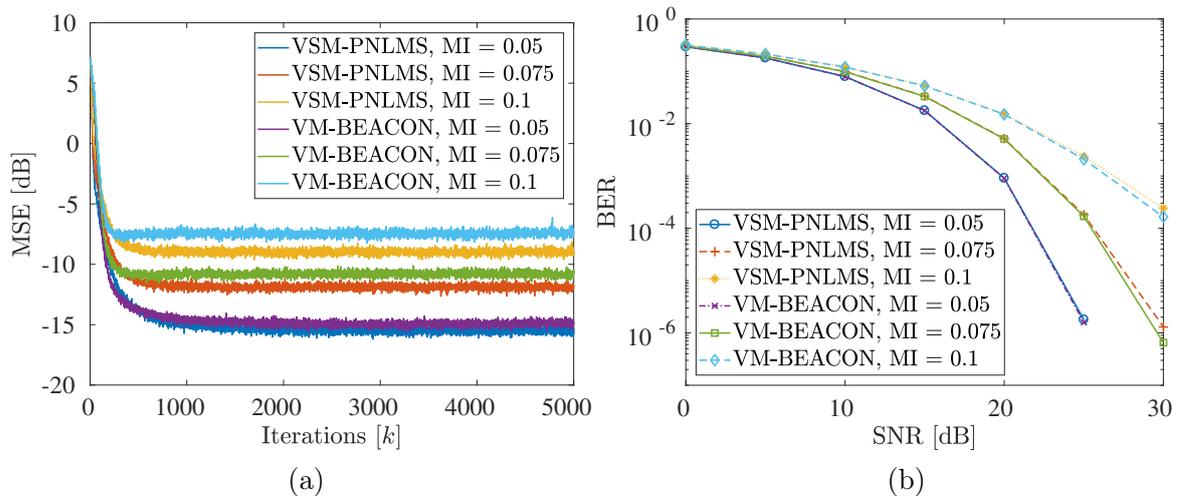


Figure 6.1: MSE (a) and BER (b) in the feedforward case for VSM-PNLMS and VM-BEACON.

Table 6.2: Iterations until steady state / Average update rates using feedforward equalization for VSM-PNLMS and VM-BEACON.

	VSM-PNLMS	VM-BEACON
MI = 0.05	717 / 12.44%	717 / 14.15%
MI = 0.075	450 / 28.44%	219 / 33.78%
MI = 0.1	332 / 44.00%	178 / 50.62%

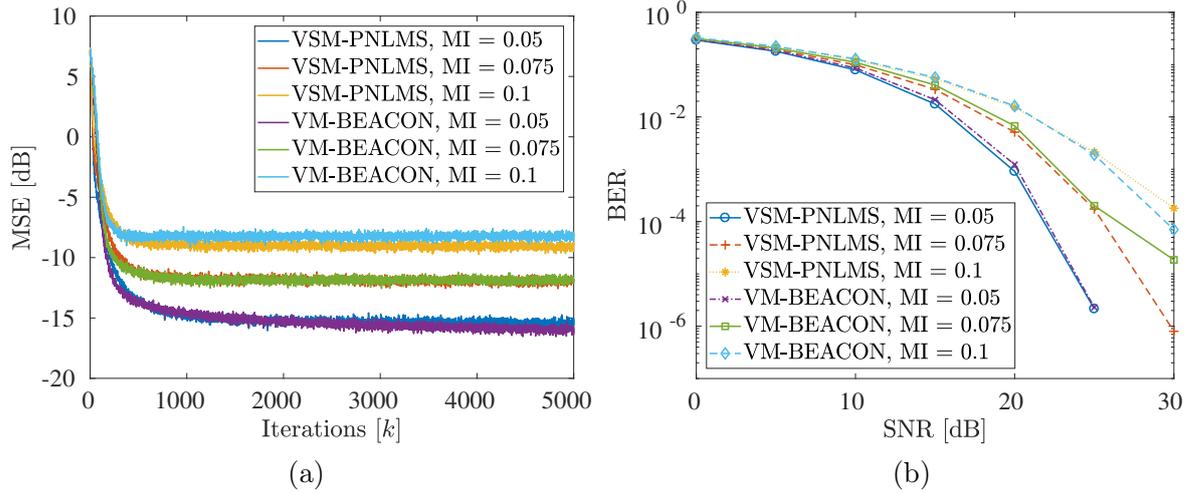


Figure 6.2: MSE (a) and BER (b) for  $M_{\text{FF}} = 8$  in the DFE case for VSM-PNLMS and VM-BEACON.

Table 6.3: Iterations until steady state / Average update rates using DFE equalization for VSM-PNLMS and VM-BEACON.

	VSM-PNLMS	VM-BEACON
MI = 0.05	771 / 13.15%	1448 / 12.32%
MI = 0.075	512 / 28.74%	438 / 29.04%
MI = 0.1	376 / 43.98%	244 / 47.69%

### 6.1.4 Conclusions of VSM-PNLMS and VM-BEACON Simulations

Considering both feedforward and DFE equalization schemes, as the modulation index increases, higher is the steady-state MSE, which is corroborated by Figures 6.1(a) and 6.2(a). In the same manner, the resulting BER for any SNR level grows as MI rises, as illustrated by Figures 6.1(b) and 6.2(b). Comparing the performance of each technique, the steady-state MSE level of VSM-PNLMS is slightly lower than VM-BEACON's for the three MI levels employed, except for MI = 0.05 in the DFE case. Moreover, the BER levels for these techniques are very similar. For MI = 0.1, which leads to a great level of nonlinearity in the VLC transceiver, both techniques provided a BER of approximately  $10^{-4}$ . Regarding the convergence speed, both methods present similar results, except for the DFE case for MI = 0.05. With respect to the update rates, VSM-PNLMS's are lower than VM-BEACON's, even with the latter presenting a computational complexity per update quadratic in relation to the filter length. As indicated in Table 6.2, VSM-PNLMS updated in 44.00% of the iterations, while VM-BEACON updated in 50.62% for MI = 0.1. Therefore, the proposed techniques reduced drastically the computational burden of Volterra series, besides presenting great equalization results considering the VLC

simulator developed in this work.

### 6.1.5 Results of VDTSM-PNLMS Algorithm

This subsection presents the equalization results for the Volterra double-threshold SM-PNLMS technique. Figures 6.3(a-c) illustrate the MSE results using different values of  $\bar{\gamma}_2$  and modulation indexes in the feedforward equalization branch, while Figure 6.4(a-c) describe the corresponding BERs. Table 6.4 displays the convergence speed and updates rates  $\nu_1$  and  $\nu_2$  for each error threshold  $\bar{\gamma}_2$  and MI, while Figure 6.8 illustrates the corresponding NUE. In the DFE case, the results are described in Figures 6.6, 6.7, 6.8 (MSE, BER, NUE, respectively), and Table 6.5.

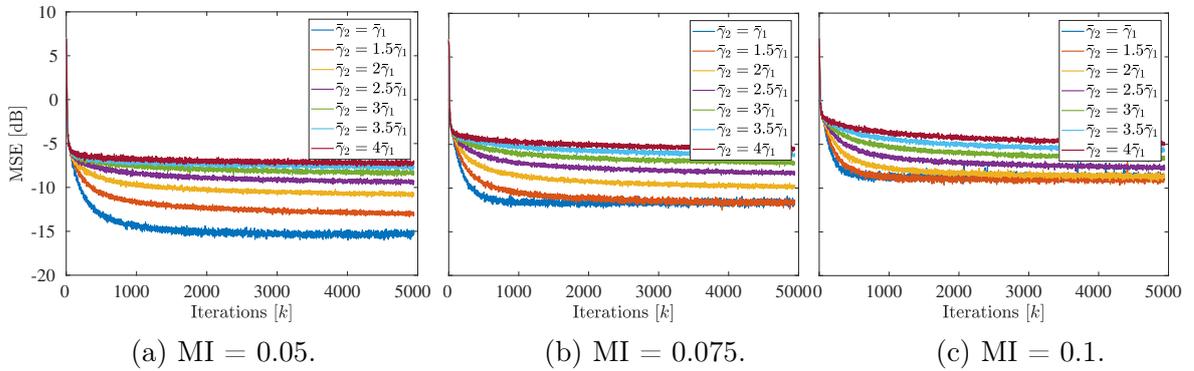


Figure 6.3: MSE for different values of  $\bar{\gamma}_2$  and MI in feedforward case for VDTSM-PNLMS.

Table 6.4: Iterations until steady state / Average update rates  $\nu_1$  / Average update rates  $\nu_2$  using feedforward equalization for VDTSM-PNLMS.

	MI = 0.05	MI = 0.075	MI = 0.1
$\bar{\gamma}_2 = \bar{\gamma}_1$	1060 / 0 / 13.82 %	459 / 0 / 29.84 %	322 / 0 / 45.55%
$\bar{\gamma}_2 = 1.5\bar{\gamma}_1$	1912 / 22.69 / 6.42 %	1460 / 21.76 / 14.41%	587 / 20.46 / 26.69%
$\bar{\gamma}_2 = 2\bar{\gamma}_1$	1663 / 43.64 / 3.82%	1945 / 48.26 / 7.85 %	1315 / 41.12 / 15.64%
$\bar{\gamma}_2 = 2.5\bar{\gamma}_1$	1590 / 54.10 / 2.50 %	2026 / 63.67 / 4.93 %	1708 / 59.40 / 9.21%
$\bar{\gamma}_2 = 3\bar{\gamma}_1$	1397 / 59.78 / 1.58 %	1930 / 70.42 / 3.39%	2116 / 68.94 / 6.17%
$\bar{\gamma}_2 = 3.5\bar{\gamma}_1$	705 / 62.70 / 0.93 %	1344 / 73.62 / 2.32%	2115 / 74.20 / 4.41%
$\bar{\gamma}_2 = 4\bar{\gamma}_1$	260 / 64.25 / 0.54 %	1410 / 75.48 / 1.53%	1897 / 77.45 / 3.15%

### 6.1.6 Conclusions of VDTSM-PNLMS Simulations

Considering the two equalization techniques, the error threshold that leads the lowest MSE level for the three employed MIs was  $\bar{\gamma}_2 = \bar{\gamma}_1$ . As shown in Figures 6.4 and 6.7, a perfect equalization was achieved for all used values of  $\bar{\gamma}_2$  and for SNR = 30 dB. Considering MI = 0.1,  $\bar{\gamma}_2 = \{2.5, 3\} \times \bar{\gamma}_1$ , were the values of error threshold which

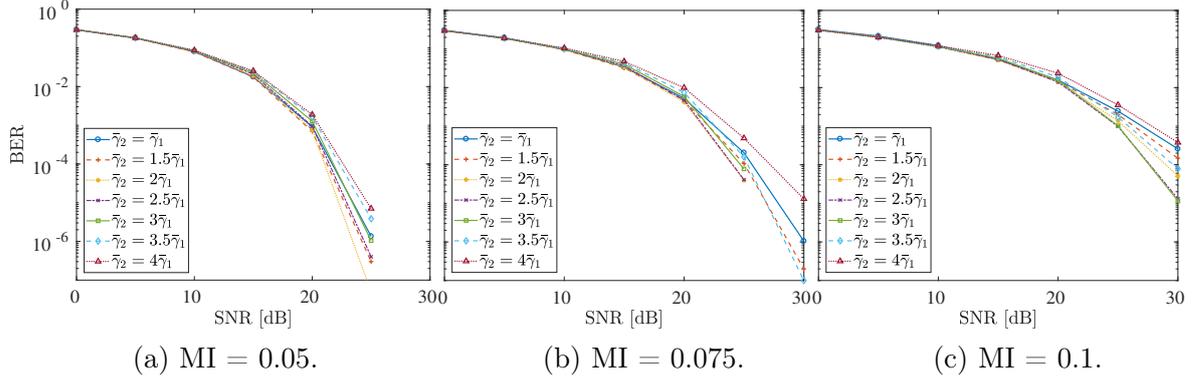


Figure 6.4: BER for different values of  $\bar{\gamma}_2$  and MI in feedforward case for VDTSM-PNLMS.

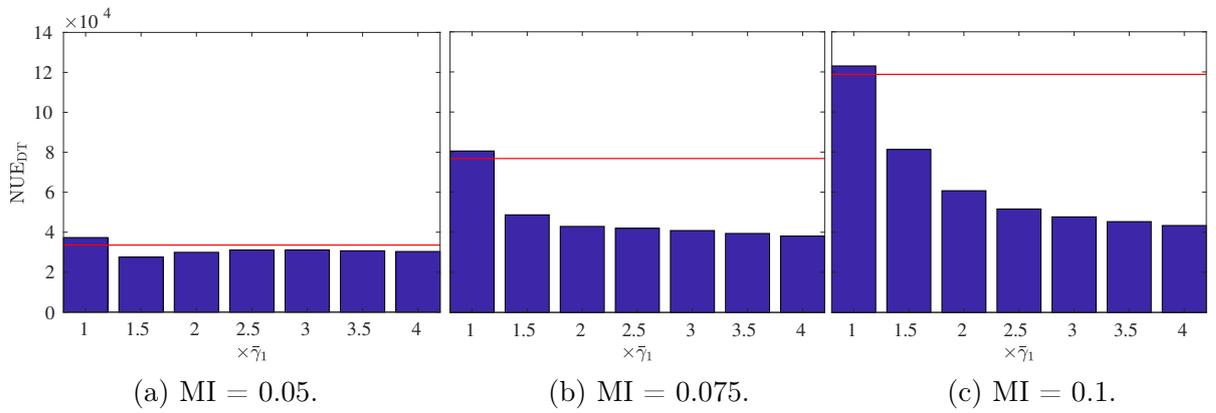


Figure 6.5: NUE for different values of  $\bar{\gamma}_2$  and MI in feedforward case for VDTSM-PNLMS. The red line indicates the NUE in the VSM-PNLMS case.

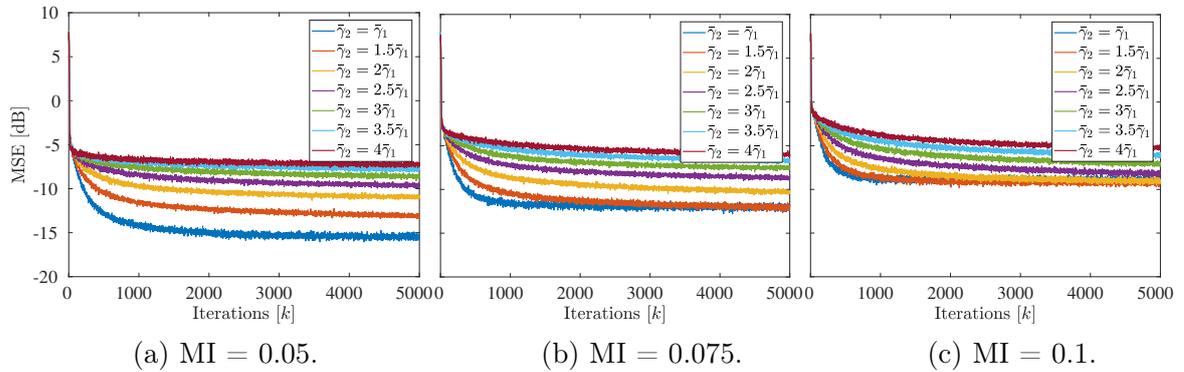


Figure 6.6: MSE for different values of  $\bar{\gamma}_2$  and MI in DFE case for VDTSM-PNLMS.

led to the lowest BER levels in both equalization scenarios, viz.:  $5.1 \times 10^{-5}$  and  $1.1 \times 10^{-5}$ , in the feedforward case, and  $9.3 \times 10^{-5}$  and  $9.8 \times 10^{-5}$  in the DFE case. Regarding the convergence speed and update rates for these error thresholds, one can infer from Figures 6.5 and 6.8 that the computational complexity is lower than in the VSM-PNLMS case. In fact, for  $\bar{\gamma}_2 = 3\bar{\gamma}_1$ , the NUE considering both equalization schemes represent only 40.14% and 42.57% of the NUE of VSM-PNLMS, even with a lower BER level, reducing even more the computational burden due to Volterra

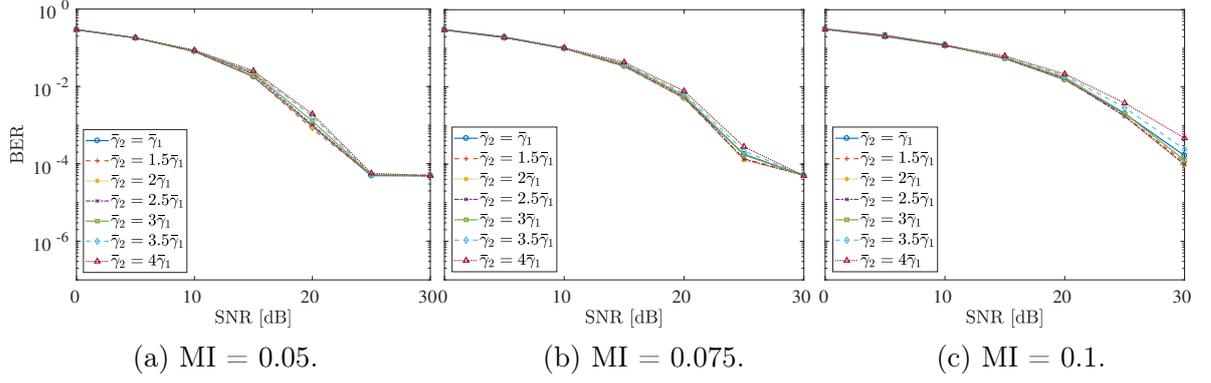


Figure 6.7: BER for different values of  $\bar{\gamma}_2$  and MI in DFE case for VDTSM-PNLMS.

Table 6.5: Iterations until steady state / Average update rates  $\nu_1$  / Average update rates  $\nu_2$  using DFE equalization for VDTSM-PNLMS.

	MI = 0.05	MI = 0.075	MI = 0.1
$\bar{\gamma}_2 = \bar{\gamma}_1$	1060 / 0 / 14.13 %	459 / 0 / 29.28 %	322 / 0 / 45.60%
$\bar{\gamma}_2 = 1.5\bar{\gamma}_1$	1912 / 22.18 / 6.83 %	1460 / 21.21 / 13.99%	587 / 20.30 / 26.49%
$\bar{\gamma}_2 = 2\bar{\gamma}_1$	1663 / 41.53 / 4.09%	1945 / 43.15 / 8.27 %	1315 / 37.42 / 15.54%
$\bar{\gamma}_2 = 2.5\bar{\gamma}_1$	1590 / 51.42 / 2.65 %	2026 / 57.86 / 5.45 %	1708 / 51.49 / 10.12%
$\bar{\gamma}_2 = 3\bar{\gamma}_1$	1397 / 57.01 / 1.71 %	1930 / 65.16 / 3.74%	2116 / 61.60 / 7.19%
$\bar{\gamma}_2 = 3.5\bar{\gamma}_1$	705 / 60.72 / 1.10 %	1344 / 69.21 / 2.65%	2115 / 68.28 / 5.25%
$\bar{\gamma}_2 = 4\bar{\gamma}_1$	260 / 62.96 / 0.64 %	1410 / 71.67 / 1.79%	1897 / 72.66 / 3.82%

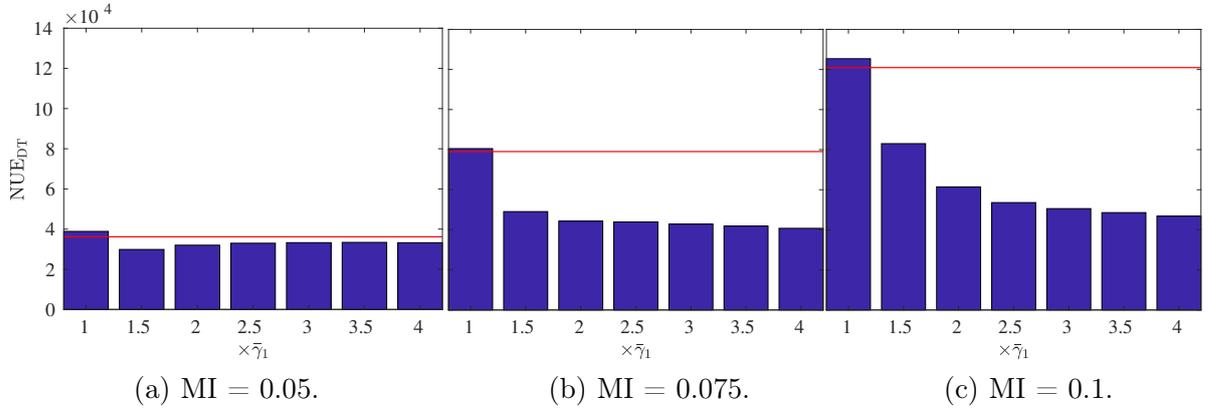


Figure 6.8: NUE for different values of  $\bar{\gamma}_2$  and MI in DFE case for VDTSM-PNLMS. The red line indicates the NUE in the VSM-PNLMS case.

series. Moreover, except for  $\bar{\gamma}_2 = \bar{\gamma}_1$ , the resulting computational complexity is lower for the remaining error threshold, with a performance in terms of BER similar to VSM-PNLMS algorithm.

### 6.1.7 Results of VSBSM-PNLMS and VSBM-BEACON Algorithms

The performance of the semi-blind techniques VSBSM-PNLMS and VSBM-BEACON is assessed in this subsection. Considering feedforward equalization, the MSE results are displayed in Figure 6.9 for the three modulation indexes employed in the simulations. Regarding the performance in terms of BER, Figure 6.10 illustrates it for each value of MI. With respect to the *Blind iteration*, i.e., the mean iteration in which the unsupervised period starts during training, and the update rates during both supervised and unsupervised periods, such results are shown in Table 6.6. All of these aforementioned results are displayed for the DFE case in Figures 6.11, 6.12 (MSE and BER), and Table 6.7.

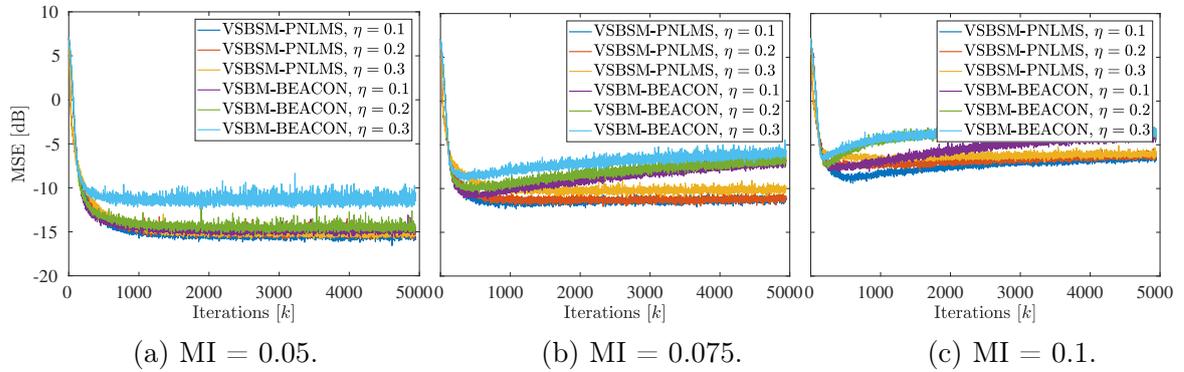


Figure 6.9: MSE for different values of MI in feedforward case for VSBSM-PNLMS and VSBM-BEACON.

Table 6.6: Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the feedforward case.

		<i>Blind iteration</i>	$\nu_{SP}$	$\nu_{BP}$	Overall update rate
MI = 0.05	$\eta = 0.1$	337 / 323	50.87 / 53.76%	10.25 / 12.07%	12.50 / 14.27%
	$\eta = 0.2$	185 / 207	67.18 / 70.71%	11.87 / 13.28%	13.24 / 14.95%
	$\eta = 0.3$	165 / 178	69.60 / 75.94%	12.34 / 19.76%	13.52 / 21.07%
MI = 0.075	$\eta = 0.1$	429 / 419	53.48 / 53.47%	27.83 / 40.45%	29.73 / 41.39%
	$\eta = 0.2$	217 / 219	70.27 / 72.00%	29.34 / 42.40%	30.61 / 43.33%
	$\eta = 0.3$	170 / 184	74.95 / 77.64%	34.09 / 48.04%	34.98 / 48.76%
MI = 0.1	$\eta = 0.1$	886 / 1827	52.16 / 54.91%	51.88 / 64.74%	51.93 / 61.23%
	$\eta = 0.2$	256 / 241	71.98 / 73.23%	54.37 / 69.95%	55.06 / 70.07%
	$\eta = 0.3$	185 / 195	78.25 / 78.65%	58.18 / 70.43%	58.67 / 70.65%

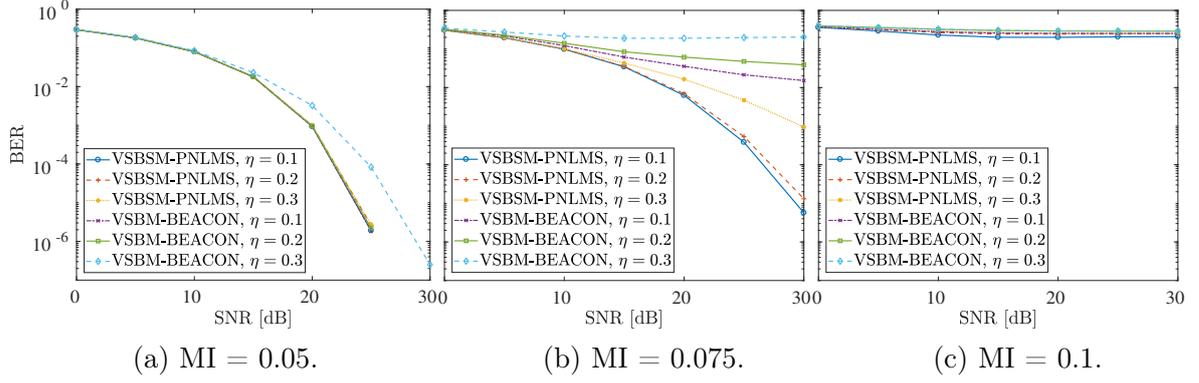


Figure 6.10: BER for different values of MI in feedforward case for VSBSM-PNLMS and VSBM-BEACON.

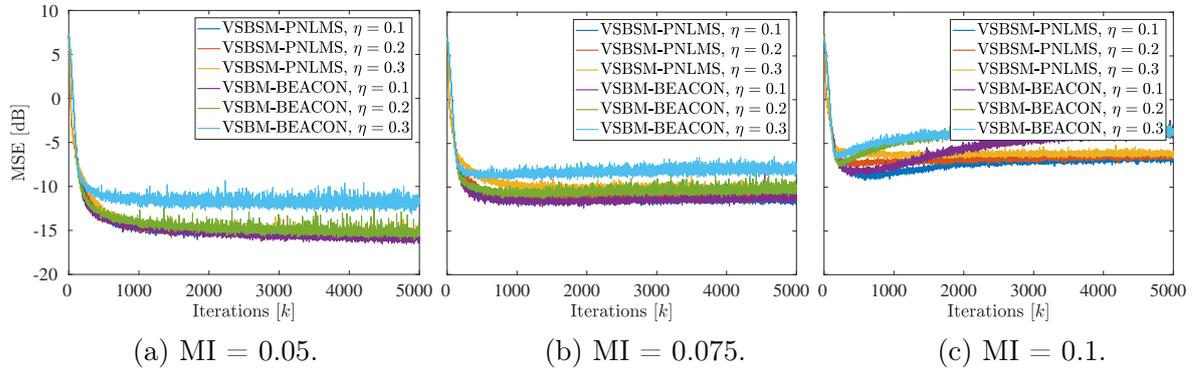


Figure 6.11: MSE for different values of MI in DFE case for VSBSM-PNLMS and VSBM-BEACON.

Table 6.7: Average number of iterations until unsupervised period and update rates for VSBSM-PNLMS / VSBM-BEACON in the DFE case.

		<i>Blind iteration</i>	$\nu_{SP}$	$\nu_{BP}$	Overall update rate
MI = 0.05	$\eta = 0.1$	353 / 327	50.91 / 53.81%	10.84 / 10.03%	13.16 / 12.33%
	$\eta = 0.2$	187 / 209	67.41 / 70.72%	12.29 / 11.50%	13.64 / 13.21%
	$\eta = 0.3$	166 / 180	70.05 / 76.24%	12.94 / 17.59%	14.11 / 18.95%
MI = 0.075	$\eta = 0.1$	455 / 420	53.38 / 53.72%	27.94 / 28.19%	29.93 / 30.01%
	$\eta = 0.2$	223 / 222	70.75 / 72.14%	29.78 / 31.29%	31.09 / 32.58%
	$\eta = 0.3$	171 / 187	75.74 / 77.53%	34.98 / 39.44%	35.86 / 40.38%
MI = 0.1	$\eta = 0.1$	941 / 1397	52.47 / 52.45%	51.81 / 65.65%	51.93 / 62.13%
	$\eta = 0.2$	268 / 245	72.44 / 73.35%	54.77 / 69.05%	55.49 / 69.20%
	$\eta = 0.3$	188 / 197	78.66 / 78.82%	58.83 / 69.98%	59.32 / 70.22%

## 6.1.8 Conclusions of VSBSM-PNLMS and VSBM-BEACON Simulations

Despite similar results with respect to MSE, feedforward and DFE equalization schemes presented distinct BER results for MI = 0.05 and MI = 0.075. Regarding feedforward equalization, for instance, for an SNR = 30 dB and MI = 0.05, the BER

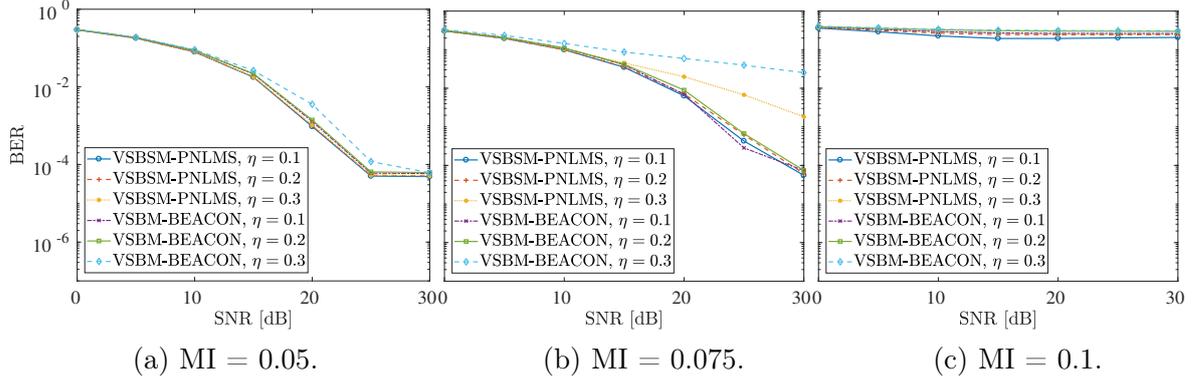


Figure 6.12: BER for different values of MI in DFE case for VSBSM-PNLMS and VSBM-BEACON.

for both algorithms was zero, except for VSBM-BEACON when using  $\eta = 0.3$ , while in the DFE case it was in the order of  $10^{-4}$ . As shown in Figures 6.10(c) and 6.12(c), none of the techniques presented satisfactory results for  $\eta = 0.3$ , whose main reason for this was that the output of the adaptive filters playing the role as reference signal was not able to keep with the learning process due to the high level of nonlinearity of the VLC transceiver. Nonetheless, for  $MI = 0.075$ , the effect of reaching a low steady-state MSE level, and this level monotonically increases, only occurred in the feedforward case for the VSBM-BEACON, which is illustrated in Figure 6.9(b), but not in the DFE fashion, as shown in Figure 6.11(b). Considering feedforward equalization, and over again for  $MI = 0.075$ , VSBSM-PNLMS yielded comparable BER levels, if compared to VSM-PNLMS', but using only a fraction of reference signal, more specifically, 429 symbols in average, as described in Table 6.6, which represents a huge saving of data and yet reducing the computational complexity due to Volterra series.

### 6.1.9 Results of VSBDTSM-PNLMS Algorithm

The following results explore the impact of the thresholds  $\eta$  and  $\bar{\gamma}_2$  in both feedforward and DFE equalization cases when using the Volterra semi-blind double-threshold SM-PNLMS algorithm for the three values of modulation index. Figures 6.13(a-c) and 6.14(a-c) show the average MSE from the 4000<sup>th</sup> up to the 5000<sup>th</sup> iteration for diverse values of  $\eta$  and  $\bar{\gamma}_2$ , and using feedforward and DFE equalizations schemes, respectively. As indicated by these results,  $\eta = 0.1$  leads to lower average MSE levels, hence, the forthcoming results consider this value. Based on the previous results and the results for this technique one can verify that, by increasing  $\eta$ , the resulting increase in MSE and BER is not proportional to the reduction of symbols employed during the training of the adaptive filters.

Considering the feedforward equalizer, the MSE and BER results are displayed in

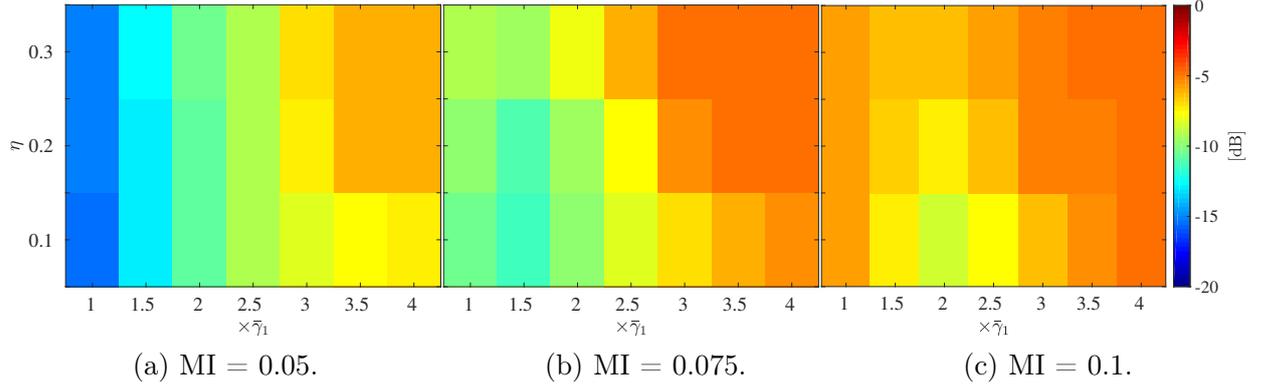


Figure 6.13: Steady-state MSE for different values of  $\bar{\gamma}_2$  and  $\eta = 0.1$  in the feedforward case for VSBDTSM-PNLMS.

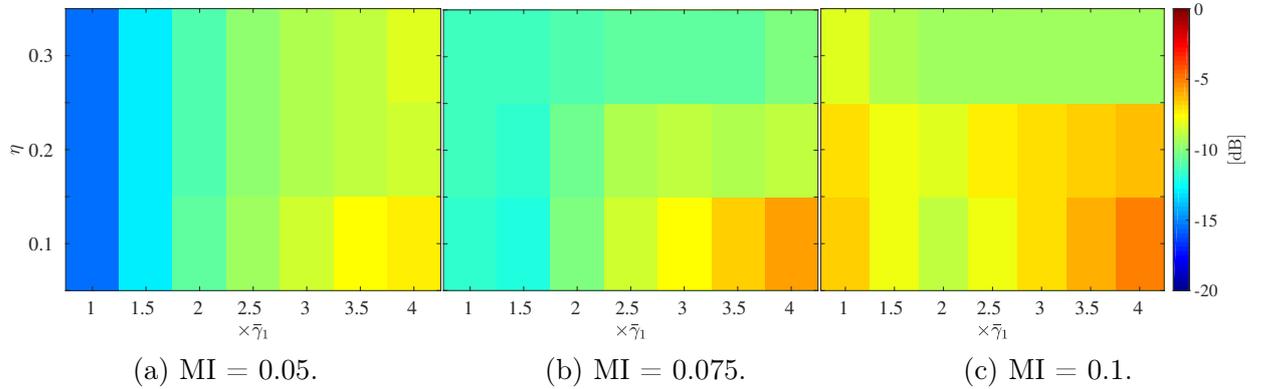


Figure 6.14: Steady-state MSE for different values of  $\bar{\gamma}_2$  and  $\eta = 0.1$  in the DFE case for VSBDTSM-PNLMS.

Figures 6.15 and 6.16, while the total updated elements is illustrated in Figure 6.17. Regarding the *Blind iteration* and update rates of the filters  $\mathbf{w}_1[k]$  and  $\mathbf{w}_2[k]$ , Table 6.8 exhibits these results. Unlike in the results described in Subsection 6.1.7, Table 6.8 does not show the update rates during the supervised and unsupervised periods, since their purpose is to indicate if the update rate during unsupervised period is much higher than in the supervised period, i.e., if the learning process continues in its natural pace.

Analogously, the MSE and BER results of VSBDTSM-PNLMS for the DFE case are shown in Figures 6.18 and 6.19, whereas Table 6.9 and Figure 6.20 describe the *Blind iteration* and updates rates, and NUE, respectively.

### 6.1.10 Conclusions of VSBDTSM-PNLMS Simulations

In general, feedforward equalization present better results in terms of BER than using DFE. For instance, if one considers MI = 0.05, in the feedforward case for all values of  $\bar{\gamma}_2$  and SNR = 30 dB, no bit errors occurred, as described in Figure 6.15(a). Besides, for MI = 0.1, which yields a severe level of nonlinearity, a BER of  $9 \times 10^{-6}$

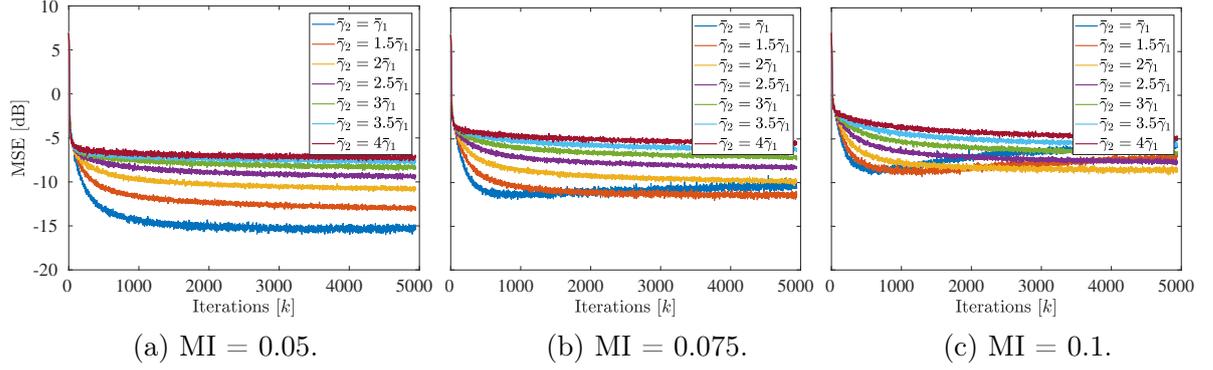


Figure 6.15: MSE for different thresholds  $\bar{\gamma}_2$  and MIs in the feedforward case for VSBDSM-PNLMS.

Table 6.8: Average number of iterations until unsupervised period / Average update rates  $\nu_1$  / Average update rates  $\nu_2$  using  $\eta = 0.1$  for VSBDSM-PNLMS in the feedforward case.

	MI = 0.05	MI = 0.075	MI = 0.1
$\gamma_2 = \gamma_1$	378 / 0 / 13.84%	496 / 0 / 31.96%	1087 / 0 / 54.92%
$\gamma_2 = 1.5\gamma_1$	1187 / 22.69 / 6.42%	1244 / 21.80 / 14.73%	1776 / 19.54 / 31.64%
$\gamma_2 = 2\gamma_1$	2122 / 43.63 / 3.82%	3283 / 48.29 / 7.88%	3736 / 41.13 / 15.81%
$\gamma_2 = 2.5\gamma_1$	2552 / 54.11 / 2.51%	1953 / 63.82 / 4.94%	- / 59.36 / 9.26%
$\gamma_2 = 3\gamma_1$	2206 / 59.78 / 1.58%	3178 / 70.39 / 3.38%	- / 68.86 / 6.11%
$\gamma_2 = 3.5\gamma_1$	3209 / 62.71 / 0.93%	- / 73.68 / 2.32%	- / 74.12 / 4.40%
$\gamma_2 = 4\gamma_1$	- / 64.25 / 0.54%	- / 75.65 / 1.53%	- / 77.33 / 3.15%

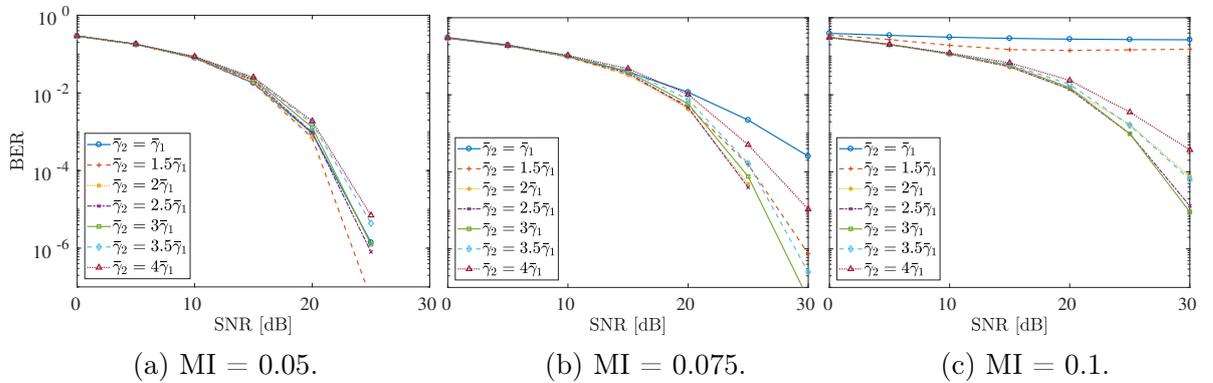


Figure 6.16: BER for different thresholds  $\bar{\gamma}_2$  and MIs in the feedforward case for VSBDSM-PNLMS.

was achieved for SNR = 30 dB by using  $\bar{\gamma}_2 = 3\bar{\gamma}_1$  (see Figure 6.15(c)). Nonetheless, the unsupervised period could not be reached due to the high steady-state MSE level, as described in Table 6.8. Among the values of  $\bar{\gamma}_2$  that could work in a semi-blind scheme, for  $\bar{\gamma}_2 = 2\bar{\gamma}_1$  yielded a BER of  $7.5 \times 10^{-5}$ , representing a huge improvement if compared to VSBSM-PNLMS results. Once again using MI = 0.1 and  $\bar{\gamma}_2 = 2\bar{\gamma}_1$ , one can observe in Figure 6.17(c) that the VSBDSM-PNLMS technique is much more efficient, in the computational complexity sense, than VSBSM-PNLMS, since

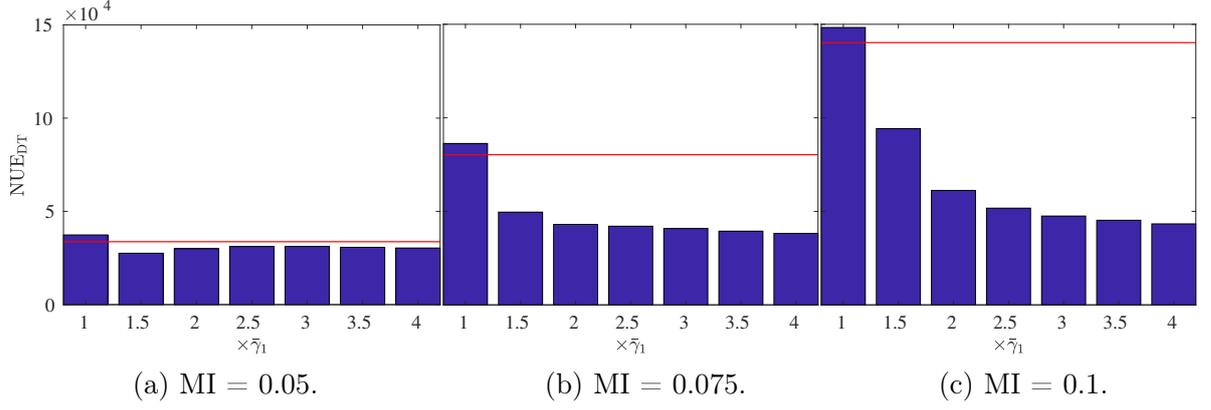


Figure 6.17: NUE for different thresholds  $\bar{\gamma}_2$  and MIs in the feedforward case for VSBDTSM-PNLMS. The red line indicates the NUE in the VSBSM-PNLMS case.

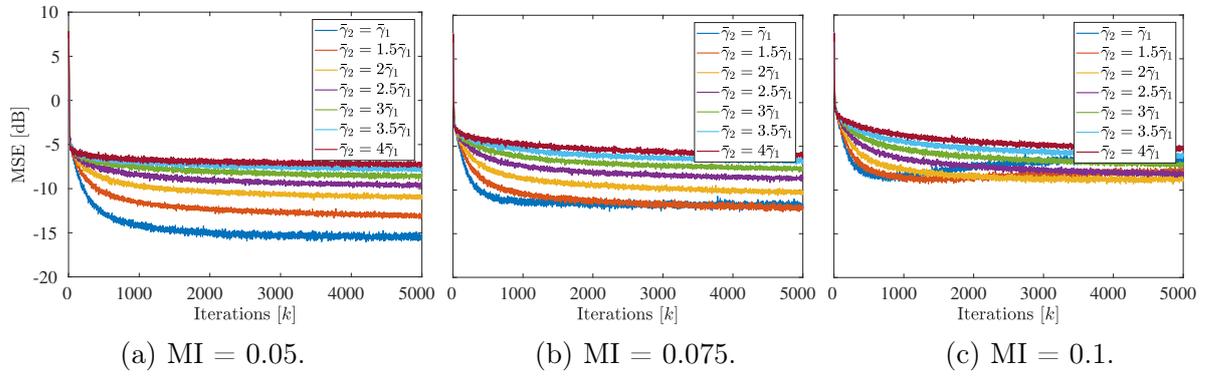


Figure 6.18: MSE for different thresholds  $\bar{\gamma}_2$  and MIs in the DFE case for VSBDTSM-PNLMS.

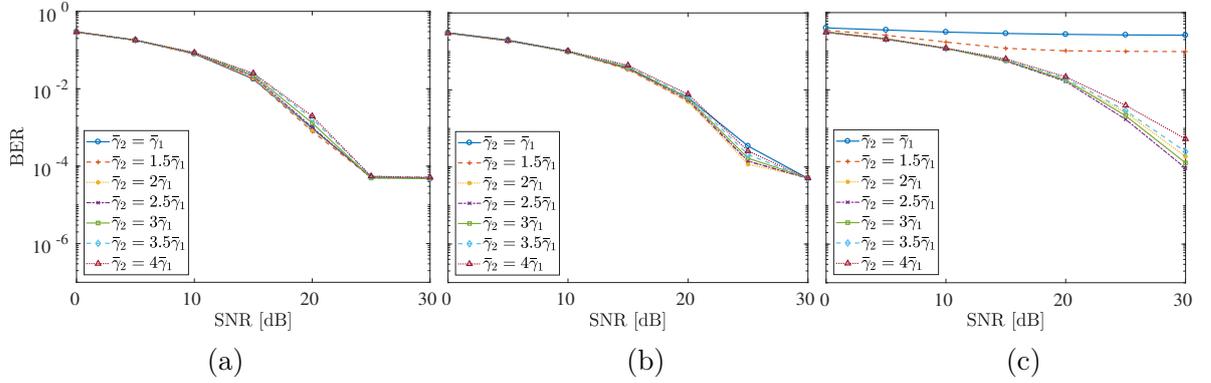


Figure 6.19: BER for different thresholds  $\bar{\gamma}_2$  and MIs in the DFE case for VSBDTSM-PNLMS.

its NUE represents only 43.64% of the latter technique, although using an amount of training data three times greater.

Table 6.9: Average number of iterations until unsupervised period / Average update rates  $\nu_1$  / Average update rates  $\nu_2$  using  $\eta = 0.1$  for VSB DTSM-PNLMS in the DFE case.

	MI = 0.05	MI = 0.075	MI = 0.1
$\gamma_2 = \gamma_1$	407 / 0 / 14.14%	544 / 0 / 29.97%	1228 / 0 / 52.09%
$\gamma_2 = 1.5\gamma_1$	1143 / 22.18 / 6.83%	1181 / 21.27 / 14.07%	1895 / 19.71 / 29.76%
$\gamma_2 = 2\gamma_1$	2251 / 41.53 / 4.09%	3568 / 43.40 / 8.22%	3544 / 37.20 / 15.85%
$\gamma_2 = 2.5\gamma_1$	2444 / 51.43 / 2.65%	3157 / 57.74 / 5.44%	4158 / 51.63 / 10.13%
$\gamma_2 = 3\gamma_1$	3074 / 57.03 / 1.73%	1919 / 65.18 / 3.75%	- / 61.57 / 7.18%
$\gamma_2 = 3.5\gamma_1$	2284 / 60.74 / 1.10%	- / 69.41 / 2.60%	- / 68.41 / 5.25%
$\gamma_2 = 4\gamma_1$	3035 / 62.96 / 0.64%	- / 71.84 / 1.80%	- / 72.81 / 3.81%

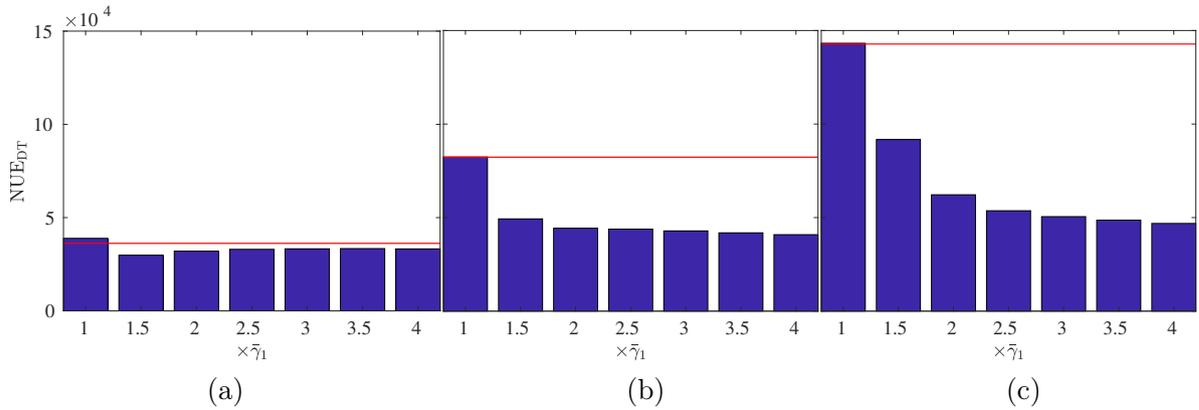


Figure 6.20: NUE for different thresholds  $\bar{\gamma}_2$  and MI in the DFE case for VSB DTSM-PNLMS. The red line indicates the NUE in the VSBSM-PNLMS case

### 6.1.11 Final Remarks

As indicated by the results of this chapter, it is possible to conclude that the proposed Volterra-based techniques are able to equalize the developed VLC transceiver considering distinct levels of nonlinearity. Due to the encouraging results presented here, these methods proved to be an interesting approach to reduce the computational burden yielded by Volterra series. Considering MI = 0.1, the lowest BER among all tested techniques was  $7.5 \times 10^{-5}$  obtained by VSB DTSM-PNLMS, with lower computational complexity, if compared with VSM-PNLMS and VSBSM-PNLMS. However, the *Blind iteration* in this case was three times larger than the VSBSM-PNLMS'. As already mentioned, it is hard to state the best algorithm in global terms. For example, a technique may provide the lowest BER at the cost of using more data as reference signal, or higher computational complexity. Nevertheless, all techniques presented in this work represent good approaches to work as equalizers in a VLC system.

# Chapter 7

## Conclusions and Future Works

### 7.1 Concluding Remarks

This work proposed data-selective techniques to be employed in a large range of nonlinear systems, whose main application is in visible light communication. One of the most popular technique to cope with the nonlinearities inherent to VLC systems is the Volterra series. Nonetheless, due to the large number of parameters to be updated in an adaptive filtering context, its use yields a high computational complexity. Therefore, by using the set-membership framework, this work proposed the use of Volterra filters with data-selective techniques, which select only data that brings innovation to the system, avoiding unnecessary updates, thereby, drastically reducing the computational burden due to Volterra series. In addition, by exploring the structure of such series, this work also proposed a technique which employs the combination of different data-selective-based adaptive filters in order to reduce even more the computational complexity. Furthermore, due to the large number of parameters associated with Volterra series, the amount of data used to train the adaptive filters may also be large. Then, this dissertation proposed the use of data-selective semi-blind techniques, which reduces both computational complexity and amount of training data.

Due to the lack of a practical tool for the performance assessment of the proposed data-selective techniques in the literature, it was also proposed a computational platform for VLC, which allows the analysis of BER under different modulation schemes or SNR levels.

Results showed that the proposed techniques are able to provide low levels of MSE and BER in both feedforward and DFE schemes, considering a generic nonlinear systems, as well as in a specific VLC system, while drastically reducing the computational burden due to Volterra series. In addition, considering semi-blind methods, a reasonable performance in terms of MSE and BER was achieved, even

using a reduced amount of training data, yet reducing the computational complexity.

It is worth mentioning that the implementation of the proposed techniques of this work can be found at [116].

## 7.2 Research Directions

As future works, the proposed techniques should be evaluated in a real scenario, including an experimental VLC system, so as to analyze the fidelity of the models of the proposed simulator. Furthermore, a deeper analysis regarding the convergence aspects of the algorithms based on different filters should be performed. Moreover, other approaches to reduce the computational complexity of Volterra series, e.g., sparse and tensor analyses, should also be investigated.

Considering the developed VLC simulator, other features, such as a model for non-line-of-sight channels, multiple LEDs and photodiodes, and other types of such devices should be incorporated to this platform, which would allow for the performance evaluation of MIMO systems. In addition, schemes that consider a nearly constant average optical power and the mitigation of flicker effects should be studied. Furthermore, techniques to address the transmission over low SNR should also be considered as future work.

# Appendix A

## Derivation of the Adaptive Algorithms

### A.1 SM-PNLMS

The cost function to be minimized in this technique is expressed as:

$$F = \frac{1}{2} \|\mathbf{w}[k+1] - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \quad (\text{A.1})$$

subject to :  $d[k] - \mathbf{w}^H[k+1]\mathbf{x}[k] = \bar{\gamma} \text{sign}\{e[k]\}$ .

Expanding (A.1), and making use of the Lagrange multiplier  $\lambda[k]$ , results in the following Lagrangian function

$$\begin{aligned} \mathcal{L}_F = & \frac{1}{2} (\mathbf{w}[k+1] - \mathbf{w}[k])^H \mathbf{G}^{-1}[k] (\mathbf{w}[k+1] - \mathbf{w}[k]) + \\ & \frac{1}{2} \lambda^*[k] (d^*[k] - \mathbf{w}^T[k+1]\mathbf{x}^*[k] - \bar{\gamma} \text{sign}\{e[k]\}) + \\ & \frac{1}{2} \lambda[k] (d[k] - \mathbf{w}^H[k+1]\mathbf{x}[k] - \bar{\gamma} \text{sign}\{e[k]\}). \end{aligned} \quad (\text{A.2})$$

Deriving (A.2) with relation to  $\mathbf{w}^*[k+1]$  and equaling to zero results in

$$\frac{\partial \mathcal{L}_F}{\partial \mathbf{w}^*[k+1]} = \mathbf{G}^{-1}[k]\mathbf{w}[k+1] - \mathbf{G}^{-1}[k]\mathbf{w}[k] - \lambda[k]\mathbf{x}[k] = 0, \quad (\text{A.3})$$

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \lambda[k]\mathbf{G}[k]\mathbf{x}[k]. \quad (\text{A.4})$$

By premultiplying (A.4) by  $\mathbf{x}^H[k]$ , gives us

$$\mathbf{x}^H[k]\mathbf{w}[k+1] = \mathbf{x}^H[k]\mathbf{w}[k] + \lambda[k]\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]. \quad (\text{A.5})$$

By using the constraint  $d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k+1] = \bar{\gamma} \text{sign}\{e[k]\}$ , (A.5) can be written

as

$$\begin{aligned}
d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k+1] &= d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k] - \lambda[k]\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k] \\
\bar{\gamma}\text{sign}\{e[k]\} &= e^*[k] - \lambda[k]\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k] \\
\lambda[k] &= \frac{e^*[k] - \bar{\gamma}\text{sign}\{e[k]\}}{\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]}.
\end{aligned} \tag{A.6}$$

By substituting (A.6) in (A.4), gives us

$$\mathbf{w}[k+1] = \begin{cases} \mathbf{w}[k] + \frac{\mathbf{G}[k]\mathbf{x}[k]}{\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]}(e^*[k] - \bar{\gamma}\text{sign}\{e[k]\}), & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.7}$$

Nonetheless, (A.7) can still be simplified further by using the fact that

$$\bar{\gamma}\text{sign}\{e[k]\} = \frac{\bar{\gamma}e^*[k]}{|e[k]|}. \tag{A.8}$$

Hence,

$$e^*[k] - \bar{\gamma}\text{sign}\{e[k]\} = e^*[k] \underbrace{\left(1 - \frac{\bar{\gamma}}{|e[k]|}\right)}_{\mu[k]}. \tag{A.9}$$

Therefore, (A.7) can be rewritten as

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu[k] \frac{\mathbf{G}[k]\mathbf{x}[k]}{\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k]} e^*[k], \tag{A.10}$$

with

$$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{|e[k]|}, & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \tag{A.11}$$

## A.2 BEACON

In this case, the cost function to be minimized is

$$\begin{aligned}
F &= \sum_{i=1}^k \lambda^{k-i}[k] |d[i] - \mathbf{w}^H[k]\mathbf{x}[i]|^2 \\
&\text{subject to : } |d[k] - \mathbf{w}^H[k]\mathbf{x}[k]|^2 = \bar{\gamma}^2.
\end{aligned} \tag{A.12}$$

Then, by applying Lagrange multipliers, it is possible to express the Lagrangian function as

$$\mathcal{L}_F = \sum_{i=0}^k \lambda^{k-i}[k] (|d[i] - \mathbf{w}^H[k]\mathbf{x}[k]|^2) + \lambda[k] (|d[i] - \mathbf{w}^H[k]\mathbf{x}[k]|^2 - \bar{\gamma}^2). \quad (\text{A.13})$$

Note that  $\lambda[k]$  in (A.12) is a parameter that can be chosen by the filter designer; in (A.13) it is set as the Lagrange multiplier associated with the Lagrangian function  $\mathcal{L}_F$ . Deriving (A.13) with relation to  $\mathbf{w}^*[k]$  and equating to zero results in

$$\frac{\partial \mathcal{L}_F}{\partial \mathbf{w}^*[k]} = \sum_{i=0}^k \lambda^{k-i}[k] (\mathbf{x}[i]\mathbf{x}^H[i]\mathbf{w}[k] - d^*[i]\mathbf{x}[i]) + \quad (\text{A.14})$$

$$\lambda[k] (\mathbf{x}[k]\mathbf{x}^H[k]\mathbf{w}[k] - d^*[k]\mathbf{x}[k]) = 0,$$

$$\mathbf{w}[k] = \underbrace{\left( \sum_{i=0}^k \lambda^{k-i}[k] \mathbf{x}[i]\mathbf{x}^H[i] + \lambda[k]\mathbf{x}[k]\mathbf{x}^H[k] \right)^{-1}}_{\mathbf{S}^{-1}[k]} \times \quad (\text{A.15})$$

$$\underbrace{\left( \sum_{i=0}^k \lambda^{k-i}[k] d^*[i]\mathbf{x}[i] + \lambda[k]d^*[k]\mathbf{x}[k] \right)}_{\mathbf{z}[k]}.$$

Therefore, (A.15) can be rewritten in a more compact fashion as

$$\mathbf{w}[k] = \mathbf{S}^{-1}[k]\mathbf{z}[k]. \quad (\text{A.16})$$

By analyzing the equation of  $\mathbf{S}[k]$  and assuming  $\lambda[k] \approx \lambda[k-1]$ , one can describe the update equation for  $\mathbf{S}[k]$  as

$$\begin{aligned} \mathbf{S}[k] &= \sum_{i=0}^k \lambda^{k-i}[k] \mathbf{x}[i]\mathbf{x}^H[i] + \lambda[k]\mathbf{x}[k]\mathbf{x}^H[k], \\ &= \sum_{i=0}^{k-1} \lambda^{k-i}[k] \mathbf{x}[i]\mathbf{x}^H[i] + \lambda[k]\mathbf{x}[k-1]\mathbf{x}^H[k-1] + \lambda[k]\mathbf{x}[k]\mathbf{x}^H[k], \\ &= \mathbf{S}[k-1] + \lambda[k]\mathbf{x}[k]\mathbf{x}^H[k]. \end{aligned} \quad (\text{A.17})$$

Analogously, the update equation for  $\mathbf{z}[k]$  can be expressed as

$$\begin{aligned}
\mathbf{z}[k] &= \sum_{i=0}^k \lambda^{k-i} [k] d^*[i] \mathbf{x}[i] + \lambda [k] d^*[k] \mathbf{x}[k], \\
&= \sum_{i=0}^{k-1} \lambda^{k-i} [k] d^*[i] \mathbf{x}[i] + \lambda [k] d^*[k-1] \mathbf{x}[k-1] + \lambda [k] d^*[k] \mathbf{x}[k], \\
&= \mathbf{z}[k-1] + \lambda [k] d^*[k] \mathbf{x}[k].
\end{aligned} \tag{A.18}$$

Then, by making use of the matrix inversion lemma [95], the update equation of  $\mathbf{S}[k]$  is described as

$$\mathbf{S}^{-1}[k] = \mathbf{S}^{-1}[k-1] - \frac{\lambda [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k] \mathbf{x}^H [k] \mathbf{S}^{-1}[k-1]}{1 + \lambda [k] \mathbf{x}^H [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k]} \tag{A.19}$$

$$= \mathbf{S}^{-1}[k-1] - \lambda [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k] \mathbf{k}^T [k], \tag{A.20}$$

where

$$\mathbf{k}[k] = \frac{\mathbf{S}^{-1}[k-1] \mathbf{x}^* [k]}{1 + \lambda [k] \mathbf{x}^H [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k]}. \tag{A.21}$$

Therefore, by using (A.18) and (A.20) in (A.16), results in

$$\begin{aligned}
\mathbf{w}[k] &= \mathbf{S}^{-1}[k] (\mathbf{z}[k-1] + \lambda [k] d^*[k] \mathbf{x}[k]), \\
&= \mathbf{S}^{-1}[k] (\mathbf{S}[k-1] \mathbf{w}[k-1] + \lambda [k] d^*[k] \mathbf{x}[k]), \\
&= \mathbf{S}^{-1}[k] \mathbf{S}[k-1] \mathbf{w}[k-1] + \mathbf{S}^{-1}[k] \lambda [k] d^*[k] \mathbf{x}[k], \\
&= (\mathbf{S}^{-1}[k-1] - \lambda [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k] \mathbf{k}^T [k]) (\mathbf{S}[k-1] \mathbf{w}[k-1]) + \mathbf{S}^{-1}[k] \lambda [k] d^*[k] \mathbf{x}[k], \\
&= \mathbf{w}[k-1] - \lambda [k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k] \mathbf{k}^T [k] \mathbf{S}[k-1] \mathbf{w}[k-1] + \mathbf{S}^{-1}[k] \lambda [k] d^*[k] \mathbf{x}[k].
\end{aligned} \tag{A.22}$$

By using (A.21) in (A.22), results in

$$\mathbf{w}[k] = \mathbf{w}[k-1] - \lambda [k] \mathbf{x}^H [k] \mathbf{w}[k-1] \mathbf{k}^* [k] + \lambda [k] d^*[k] \mathbf{S}^{-1}[k] \mathbf{x}[k]. \tag{A.23}$$

Using the fact that  $\mathbf{k}^* [k] = \mathbf{S}^{-1}[k] \mathbf{x}[k]$ , the previous equation is rewritten as

$$\begin{aligned}
\mathbf{w}[k] &= \mathbf{w}[k-1] + \lambda [k] (-\mathbf{x}^H \mathbf{w}[k-1] \mathbf{k}[k] + d^*[k] \mathbf{S}^{-1}[k] \mathbf{x}[k]), \\
&= \mathbf{w}[k-1] + \lambda [k] (-\mathbf{x}^H \mathbf{w}[k-1] + d^*[k]) \mathbf{k}^* [k], \\
&= \mathbf{w}[k-1] + \lambda [k] e^*[k] \mathbf{k}^* [k], \\
&= \mathbf{w}[k-1] + \lambda [k] \mathbf{S}^{-1}[k] \mathbf{x}[k] e^*[k],
\end{aligned} \tag{A.24}$$

where the *a priori* error is defined here as  $e[k] = d[k] - \mathbf{w}^H [k-1] \mathbf{x}[k]$ . Now, it

is only left the evaluation of  $\lambda[k]$ . To do that, let us premultiply (A.24) by  $\mathbf{x}^H[k]$ , which leads

$$\mathbf{x}^H[k]\mathbf{w}[k] = \mathbf{x}^H[k]\mathbf{w}[k-1] + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k]\mathbf{x}[k]e^*[k]. \quad (\text{A.25})$$

Then, by subtracting  $d^*[k]$  of the previous equation, it can be rewritten as

$$d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k] = e^*[k] - \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k]\mathbf{x}[k]e^*[k], \quad (\text{A.26})$$

which can be rewritten once again by using (A.19) as

$$\begin{aligned} d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k] &= e^*[k] - \lambda[k]\mathbf{x}^H[k] \times \\ &\left( \mathbf{S}^{-1}[k-1] - \frac{\lambda[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right) \mathbf{x}[k]e^*[k], \end{aligned} \quad (\text{A.27})$$

After some algebraic manipulation, (A.27) is simplified and rewritten as

$$\begin{aligned} d^*[k] - \mathbf{x}^H[k]\mathbf{w}[k] &= e^*[k] - \left( \frac{\lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right) \mathbf{x}[k]e^*[k], \\ &= e^*[k] \left( 1 - \frac{\lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right), \\ &= e^*[k] \left( \frac{1}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right). \end{aligned} \quad (\text{A.28})$$

Thus, by using the constraint of (A.12) and using the fact that  $\mathbf{S}^{-1}[k-1]$  is a real and positive-definite matrix,  $\lambda[k]$  is expressed mathematically as

$$\begin{aligned} \bar{\gamma} &= \left| e^*[k] \left( \frac{1}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right) \right|, \\ &= |e^*[k]| \left( \frac{1}{1 + \lambda[k]\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \right), \end{aligned} \quad (\text{A.29})$$

$$\begin{aligned} \lambda[k] &= \frac{1}{\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \left( \frac{|e^*[k]|}{\bar{\gamma}} - 1 \right), \\ &= \frac{1}{\mathbf{x}^H[k]\mathbf{S}^{-1}[k-1]\mathbf{x}[k]} \left( \frac{|e[k]|}{\bar{\gamma}} - 1 \right). \end{aligned} \quad (\text{A.30})$$

### A.3 M-BEACON

The derivation of this algorithm is very similar to BEACON's. In order to present its derivation, let us first define the following variables:

$$\tilde{\mathbf{S}}[k] = \lambda^{-1}[k]\mathbf{S}[k], \quad (\text{A.31})$$

$$\tilde{\mathbf{z}}[k] = \lambda^{-1}[k]\mathbf{z}[k], \quad (\text{A.32})$$

$$l[k] = \frac{\lambda[k-1]}{\lambda[k]}. \quad (\text{A.33})$$

Therefore, by multiplying both sides of (A.17) and (A.18) by  $\lambda^{-1}[k]$ , and using this previous definitions, one can rewrite these equations as

$$\tilde{\mathbf{S}}[k] = l[k]\tilde{\mathbf{S}}[k-1] + \mathbf{x}[k]\mathbf{x}^H[k], \quad (\text{A.34})$$

$$\tilde{\mathbf{z}}[k] = l[k]\tilde{\mathbf{z}}[k-1] + d^*[k]\mathbf{x}[k]. \quad (\text{A.35})$$

Once again employing the matrix inversion lemma yields

$$\tilde{\mathbf{S}}^{-1}[k] = l^{-1}[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \frac{\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]\mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]}{l[k] + \mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]} \right), \quad (\text{A.36})$$

$$= l^{-1}[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]\mathbf{k}^T[k] \right), \quad (\text{A.37})$$

where  $\mathbf{k}[k]$  is here redefined as

$$\mathbf{k}[k] = \frac{\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}^*[k]}{l[k] + \mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]}. \quad (\text{A.38})$$

Therefore, by using (A.35) and (A.37) in (A.16), results in

$$\begin{aligned} \mathbf{w}[k] &= \tilde{\mathbf{S}}^{-1}[k] (\tilde{\mathbf{z}}[k-1]l[k] + d^*[k]\mathbf{x}[k]), \\ &= \tilde{\mathbf{S}}^{-1}[k] \left( \tilde{\mathbf{S}}[k-1]\mathbf{w}[k-1]l[k] + d^*[k]\mathbf{x}[k] \right), \\ &= \tilde{\mathbf{S}}^{-1}[k]\tilde{\mathbf{S}}[k-1]\mathbf{w}[k-1]l[k] + \tilde{\mathbf{S}}^{-1}[k]d^*[k]\mathbf{x}[k], \\ &= l^{-1}[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]\mathbf{k}^T[k] \right) \left( \tilde{\mathbf{S}}[k-1]\mathbf{w}[k-1]l[k] \right) + \\ &\quad \tilde{\mathbf{S}}^{-1}[k]d^*[k]\mathbf{x}[k], \\ &= \mathbf{w}[k-1] - \tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]\mathbf{k}^T[k]\tilde{\mathbf{S}}[k-1]\mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k]d^*[k]\mathbf{x}[k]. \end{aligned} \quad (\text{A.39})$$

By using (A.38) in (A.39), results in

$$\mathbf{w}[k] = \mathbf{w}[k-1] - \mathbf{x}^H[k]\mathbf{w}[k-1]\mathbf{k}^*[k] + d^*[k]\tilde{\mathbf{S}}^{-1}[k]\mathbf{x}[k]. \quad (\text{A.40})$$

Over again using the fact that  $\mathbf{k}^*[k] = \tilde{\mathbf{S}}^{-1}[k]\mathbf{x}[k]$ , the previous equation is rewritten as

$$\begin{aligned}
\mathbf{w}[k] &= \mathbf{w}[k-1] + \left( -\mathbf{x}^H \mathbf{w}[k-1] \mathbf{k}[k] + d^*[k] \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}[k] \right), \\
&= \mathbf{w}[k-1] + \left( -\mathbf{x}^H \mathbf{w}[k-1] + d^*[k] \right) \mathbf{k}^*[k], \\
&= \mathbf{w}[k-1] + e^*[k] \mathbf{k}^*[k], \\
&= \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k] \mathbf{x}[k] e^*[k],
\end{aligned} \tag{A.41}$$

In order to assess  $l[k]$ , one can substitute (A.30) in (A.33), yielding

$$l[k] = \frac{\lambda[k-1] \mathbf{x}^H[k] \mathbf{S}^{-1}[k-1] \mathbf{x}[k]}{\left( \frac{|e[k]|}{\bar{\gamma}} - 1 \right)}. \tag{A.42}$$

Once  $\lambda[k-1] \mathbf{S}^{-1}[k-1] = l^{-1}[k-1] \tilde{\mathbf{S}}[k-1]$ , (A.42) can be rewritten as

$$l[k] = \frac{\mathbf{x}^H[k] \tilde{\mathbf{S}}^{-1}[k-1] \mathbf{x}[k]}{\left( \frac{|e[k]|}{\bar{\gamma}} - 1 \right)}. \tag{A.43}$$

## A.4 Volterra Semi-blind SM-PNLMS

During the supervised period, the cost function to be minimized in this case is described as

$$\begin{aligned}
F &= \frac{1}{2} \|\mathbf{w}[k-1] - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \\
&\text{subject to : } \|d[k] - \mathbf{w}^H[k+1] \mathbf{x}_N[k] - \bar{\gamma} \text{sign}_{\infty}\{e[k]\}\|_{\infty} = 0.
\end{aligned} \tag{A.44}$$

Expanding (A.44) and making use of the Lagrange multiplier  $\lambda[k]$ , results in the following Lagrangian function

$$\begin{aligned}
\mathcal{L}_F &= \frac{1}{2} (\mathbf{w}[k+1] - \mathbf{w}[k])^H \mathbf{G}^{-1}[k] (\mathbf{w}[k+1] - \mathbf{w}[k]) + \\
&\quad \frac{1}{2} \lambda^*[k] \|d^*[k] - \mathbf{w}^T[k] \mathbf{x}_N^*[k] - \bar{\gamma} \text{sign}_{\infty}\{e[k]\}\|_{\infty} + \\
&\quad \frac{1}{2} \lambda[k] \|d[k] - \mathbf{w}^H[k] \mathbf{x}_N[k] - \bar{\gamma} \text{sign}_{\infty}\{e[k]\}\|_{\infty}.
\end{aligned} \tag{A.45}$$

Defining as  $a[k] = d[k] - \mathbf{w}^H[k+1] \mathbf{x}_N[k] - \bar{\gamma} \text{sign}_{\infty}\{e[k]\}$ , one can derive (A.45)

with relation to  $\mathbf{w}^*[k+1]$  as follows

$$\begin{aligned}\frac{\partial \mathcal{L}_F}{\partial \mathbf{w}^*[k+1]} &= \mathbf{G}^{-1}[k]\mathbf{w}[k+1] - \mathbf{G}^{-1}[k]\mathbf{w}[k] - \lambda[k] \frac{\partial \|a[k]\|_\infty}{\partial a[k]} \frac{\partial a[k]}{\partial \mathbf{w}^*[k+1]}, \\ &= \mathbf{G}^{-1}[k]\mathbf{w}[k+1] - \mathbf{G}^{-1}[k]\mathbf{w}[k] - \lambda[k] \frac{\partial \|a[k]\|_\infty}{\partial a[k]} \mathbf{x}_N[k],\end{aligned}\quad (\text{A.46})$$

Defining as  $b[k] = \lambda[k] \frac{\partial \|a[k]\|_\infty}{\partial a[k]}$  and by setting (A.46) to zero, leads to

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mathbf{G}[k]\mathbf{x}_N[k]b[k]. \quad (\text{A.47})$$

By premultiplying (A.47) by  $\mathbf{x}_N^H[k]$ , subtracting  $d^*[k]$  and  $\bar{\gamma}\text{sign}_\infty\{e[k]\}$  from both sides of this equation, results in

$$\begin{aligned}0 &= d^*[k] - \mathbf{x}_N^H[k]\mathbf{w}[k] - \bar{\gamma}\text{sign}_\infty\{e[k]\} - \mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]b[k] \\ b[k] &= \frac{e^*[k] - \bar{\gamma}\text{sign}_\infty\{e[k]\}}{\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]}.\end{aligned}\quad (\text{A.48})$$

Hence, by substituting (A.48) in (A.47) leads to

$$\mathbf{w}[k+1] = \begin{cases} \mathbf{w}[k] + \frac{\mathbf{G}[k]\mathbf{x}_N[k]}{\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]}(e^*[k] - \bar{\gamma}\text{sign}_\infty\{e[k]\}), & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.49})$$

Nonetheless, (A.49) can still be simplified further by using the fact that

$$\bar{\gamma}\text{sign}_\infty\{e[k]\} = \frac{\bar{\gamma}e^*[k]}{\|e[k]\|_\infty}. \quad (\text{A.50})$$

Hence,

$$e^*[k] - \bar{\gamma}\text{sign}_\infty\{e[k]\} = e^*[k] \underbrace{\left(1 - \frac{\bar{\gamma}}{\|e[k]\|_\infty}\right)}_{\mu[k]}. \quad (\text{A.51})$$

Therefore, (A.10) can be rewritten as

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu[k] \frac{\mathbf{G}[k]\mathbf{x}_N[k]}{\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k]} e^*[k], \quad (\text{A.52})$$

with

$$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{\|e[k]\|_\infty}, & \text{if } \|e[k]\|_\infty > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.53})$$

During the unsupervised period, the only modification to be made is to replace the reference signal  $d[k]$  by the decided symbol  $b[k]$ . For this case, the error is defined as

$$e_b[k] = b[k] - \mathbf{w}^H[k] \mathbf{x}_N[k]. \quad (\text{A.54})$$

Thus, (A.44) is modified as follows:

$$F = \frac{1}{2} \|\mathbf{w}[k-1] - \mathbf{w}[k]\|_{\mathbf{G}^{-1}[k]}^2 \quad (\text{A.55})$$

subject to :  $\|b[k] - \mathbf{w}^H[k+1] \mathbf{x}_N[k] - \bar{\gamma} \text{sign}_\infty\{e_b[k]\}\|_\infty = 0.$

Following the same steps that yielded (A.10), it is possible to describe the update equation in the unsupervised period as

$$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu_b[k] \frac{\mathbf{G}[k] \mathbf{x}_N[k]}{\mathbf{x}_N^H[k] \mathbf{G}[k] \mathbf{x}_N[k]} e_b^*[k], \quad (\text{A.56})$$

with

$$\mu_b[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{\|e_b[k]\|_\infty}, & \text{if } \|e_b[k]\|_\infty > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.57})$$

# Appendix B

## Pseudo-codes of the Adaptive Algorithms

This appendix describes the adaptive algorithms in a form of pseudo-code.

### B.1 SM-PNLMS

---

**Algorithm 1:** Pseudocode of SM-PNLMS.

---

Initialization:

choose  $\bar{\gamma}$  and  $\kappa$

set  $\delta \ll 1$

$\mathbf{w}[0] = [0 \cdots 0]^T + \delta$

**for**  $k \geq 0$  **do**

$e[k] = d[k] - \mathbf{w}^H[k]\mathbf{x}[k]$

$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{|e[k]|}, & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases}$

**for**  $l = 1$  **to**  $L$  **do**

$\mathbf{G}_l[k] = \frac{1 - \kappa\mu[k]}{L} + \frac{\kappa\mu[k]|\mathbf{w}_l[k]|}{\|\mathbf{w}[k]\|_1}$

**end**

$\mathbf{w}[k + 1] = \mathbf{w}[k] + \mu[k]\mathbf{G}[k]\mathbf{x}[k](\mathbf{x}^H[k]\mathbf{G}[k]\mathbf{x}[k])^{-1}e^*[k]$

**end**

---

## B.2 M-BEACON

---

**Algorithm 2:** Pseudocode of M-BEACON.

---

Initialization:

set  $\delta \ll 1$

$\mathbf{w}[0] = [0 \cdots 0]^T$

$\tilde{\mathbf{S}}^{-1}[0] = \delta \mathbf{I}$

**for**  $k \geq 1$  **do**

$e[k] = d[k] - \mathbf{w}^H[k-1]\mathbf{x}[k]$

$\lambda[k] = \begin{cases} \mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k] \frac{1}{\left(\frac{|e[k]|}{\bar{\gamma}} - 1\right)}, & \text{if } |e[k]| > \bar{\gamma}, \\ 0, & \text{otherwise,} \end{cases}$

$\tilde{\mathbf{S}}^{-1}[k] = \lambda[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \frac{\tilde{\mathbf{S}}^{-1}[k]\mathbf{x}[k]\mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]}{\lambda^{-1}[k] + \mathbf{x}^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}[k]} \right),$

$\mathbf{w}[k] = \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k]\mathbf{x}[k]e^*[k],$

**end**

---

## B.3 VSBSM-PNLMS

---

**Algorithm 3:** Pseudocode of VSBSM-PNLMS.

---

Initialization:

choose  $\bar{\gamma}$  and  $\kappa$

set  $\delta \ll 1$

set *Blind Iteration*

$\mathbf{w}[0] = [0 \cdots 0]^T + \delta$

**for**  $k \geq 0$  **do**

build input vector  $\mathbf{x}_N[k]$  by using (3.19) and (3.20)

$y[k] = \mathbf{w}^H[k]\mathbf{x}_N[k]$

**if**  $k < \textit{Blind Iteration}$  **then**

$e[k] = d[k] - y[k]$

**else**

$b[k] = \text{dec}\{y[k]\}$

$e[k] = b[k] - y[k]$

**end**

$\mu[k] = \begin{cases} 1 - \frac{\bar{\gamma}}{\|e[k]\|_\infty}, & \text{if } \|e[k]\|_\infty > \bar{\gamma}, \\ 0, & \text{otherwise.} \end{cases}$

**for**  $l = 1$  **to**  $L$  **do**

$\mathbf{G}_l[k] = \frac{1 - \kappa\mu[k]}{L} + \frac{\kappa\mu[k]\|\mathbf{w}_l[k]\|}{\|\mathbf{w}[k]\|_1}$

**end**

$\mathbf{w}[k+1] = \mathbf{w}[k] + \mu[k]\mathbf{G}[k]\mathbf{x}_N[k](\mathbf{x}_N^H[k]\mathbf{G}[k]\mathbf{x}_N[k])^{-1}e^*[k]$

**end**

---

Note that in the previous pseudo-code,  $e_b[k]$  was changed to  $e[k]$  in order to avoid code repetition.

## B.4 VSBM-BEACON

---

**Algorithm 4:** Pseudocode of VSBM-BEACON.

---

Initialization:

set  $\delta \ll 1$

set *Blind Iteration*

$\mathbf{w}[0] = [0 \cdots 0]^T$

$\tilde{\mathbf{S}}^{-1}[0] = \delta \mathbf{I}$

**for**  $k \geq 1$  **do**

    build input vector  $\mathbf{x}_N[k]$  by using (3.19) and (3.20)

$y[k] = \mathbf{w}^H[k-1]\mathbf{x}_N[k]$

**if**  $k < \textit{Blind Iteration}$  **then**

$e[k] = d[k] - y[k]$

**else**

$b[k] = \text{dec}\{y[k]\}$

$e[k] = b[k] - y[k]$

**end**

$$\lambda[k] = \begin{cases} \mathbf{x}_N^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}_N[k] \frac{1}{\left(\frac{\|e[k]\|_\infty}{\bar{\gamma}} - 1\right)}, & \text{if } \|e[k]\|_\infty > \bar{\gamma}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\tilde{\mathbf{S}}^{-1}[k] = \lambda[k] \left( \tilde{\mathbf{S}}^{-1}[k-1] - \frac{\tilde{\mathbf{S}}^{-1}[k]\mathbf{x}_N[k]\mathbf{x}_N^H[k]\tilde{\mathbf{S}}^{-1}[k-1]}{\lambda^{-1}[k] + \mathbf{x}_N^H[k]\tilde{\mathbf{S}}^{-1}[k-1]\mathbf{x}_N[k]} \right),$$

$\mathbf{w}[k] = \mathbf{w}[k-1] + \tilde{\mathbf{S}}^{-1}[k]\mathbf{x}_N[k]e^*[k],$

**end**

---

## B.5 VDTSM-PNLMS

---

**Algorithm 5:** Pseudocode of VDTSM-PNLMS.

---

Initialization:

choose  $\bar{\gamma}_1, \bar{\gamma}_2, \kappa_1, \kappa_2,$  and  $\sigma$

set  $C = 2$

set  $\delta \ll 1$

$\mathbf{w}_1[0] = [0 \cdots 0]^T + \delta$

$\mathbf{w}_2[0] = [0 \cdots 0]^T + \delta$

$\mathbf{w}[0] = [0 \cdots 0]^T$

**for**  $k \geq 0$  **do**

    build input vector  $\mathbf{x}_N[k]$  by using (3.19) and (3.20)

$e[k] = d[k] - \mathbf{w}^H[k]\mathbf{x}_N[k]$

**for**  $c = 1$  **to**  $C$  **do**

$$\mu_c[k] = \begin{cases} 1 - \frac{\bar{\gamma}_c}{|e[k]|}, & \text{if } |e[k]| > \bar{\gamma}_c, \\ 0, & \text{otherwise.} \end{cases}$$

**for**  $l = 1$  **to**  $L$  **do**

$$\mathbf{G}_{c,l}[k] = \frac{1 - \kappa_c \mu_c[k]}{L} + \frac{\kappa_c \mu_c[k] \|\mathbf{w}_{c,l}[k]\|}{\|\mathbf{w}_c[k]\|_1}$$

**end**

$$\mathbf{w}_c[k+1] = \mathbf{w}_c[k] + \mu_c[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k] (\mathbf{x}_N^H[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k])^{-1} e^*[k]$$

**end**

$\mathbf{w}[k+1] = \sigma \mathbf{w}_1[k+1] + (1 - \sigma) \mathbf{w}_2[k+1]$  #Perform combination of  
parameter vectors

$\mathbf{w}_1[k+1] \leftarrow \mathbf{w}[k+1]$  # Assign  $\mathbf{w}[k+1]$  to  $\mathbf{w}_1[k+1]$

$\mathbf{w}_2[k+1] \leftarrow \mathbf{w}[k+1]$  # Assign  $\mathbf{w}[k+1]$  to  $\mathbf{w}_2[k+1]$

**end**

---

## B.6 VSBDTSM-PNLMS

---

**Algorithm 6:** Pseudocode of VSBDTSM-PNLMS.

---

Initialization:

choose  $\bar{\gamma}_1, \bar{\gamma}_2, \kappa_1, \kappa_2,$  and  $\sigma$

set  $C = 2$

set  $\delta \ll 1$

set *Blind Iteration*

$\mathbf{w}_1[0] = [0 \cdots 0]^T + \delta$

$\mathbf{w}_2[0] = [0 \cdots 0]^T + \delta$

$\mathbf{w}[0] = [0 \cdots 0]^T$

**for**  $k \geq 0$  **do**

    build input vector  $\mathbf{x}_N[k]$  by using (3.19) and (3.20)

$y[k] = \mathbf{w}^H[k-1]\mathbf{x}_N[k]$

**if**  $k < \textit{Blind Iteration}$  **then**

$e[k] = d[k] - y[k]$

**else**

$b[k] = \text{dec}\{y[k]\}$

$e[k] = b[k] - y[k]$

**end**

**for**  $c = 1$  **to**  $C$  **do**

$$\mu_c[k] = \begin{cases} 1 - \frac{\bar{\gamma}_c}{\|e[k]\|_\infty}, & \text{if } \|e[k]\|_\infty > \bar{\gamma}_c, \\ 0, & \text{otherwise.} \end{cases}$$

**for**  $l = 1$  **to**  $L$  **do**

$$\mathbf{G}_{c,ul}[k] = \frac{1 - \kappa_c \mu_c[k]}{L} + \frac{\kappa_c \mu_c[k] \|\mathbf{w}_{c,l}[k]\|}{\|\mathbf{w}_c[k]\|_1}$$

**end**

$$\mathbf{w}_c[k+1] = \mathbf{w}_c[k] + \mu_c[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k] (\mathbf{x}_N^H[k] \mathbf{U}_c \mathbf{G}_c[k] \mathbf{x}_N[k])^{-1} e^*[k]$$

**end**

$$\mathbf{w}[k+1] = \sigma \mathbf{w}_1[k+1] + (1 - \sigma) \mathbf{w}_2[k+1]$$
 #Perform combination of parameter vectors

$\mathbf{w}_1[k+1] \leftarrow \mathbf{w}[k+1]$  # Assign  $\mathbf{w}[k+1]$  to  $\mathbf{w}_1[k+1]$

$\mathbf{w}_2[k+1] \leftarrow \mathbf{w}[k+1]$  # Assign  $\mathbf{w}[k+1]$  to  $\mathbf{w}_2[k+1]$

**end**

---

Once again  $e_b[k]$  was changed to  $e[k]$  in order to avoid code repetition.

# Bibliography

- [1] Available at: [https://commons.wikimedia.org/wiki/File:Chappe\\_semaphore.jpg](https://commons.wikimedia.org/wiki/File:Chappe_semaphore.jpg). Online, accessed 02-20-2018. Public access.
- [2] . Available at: [https://commons.wikimedia.org/wiki/File:Photophone\\_transmitter\\_4074931746\\_9f996df841\\_b.jpg](https://commons.wikimedia.org/wiki/File:Photophone_transmitter_4074931746_9f996df841_b.jpg). Online, accessed 02-20-2018. Public access.
- [3] . Available at: [https://commons.wikimedia.org/wiki/File:Photophone\\_receiver\\_4074172975\\_288f2808f0\\_o.jpg](https://commons.wikimedia.org/wiki/File:Photophone_receiver_4074172975_288f2808f0_o.jpg). Online, accessed 02-20-2018. Public access.
- [4] WU, S., WANG, H., YOUN, C. H. “Visible light communications for 5G wireless networking systems: from fixed to mobile communications”, *IEEE Network*, v. 28, n. 6, pp. 41–45, Nov 2014.
- [5] HAAS, H., YIN, L., WANG, Y., et al. “What is LiFi?” *Journal of Lightwave Technology*, v. 34, n. 6, pp. 1533–1544, Mar 2016.
- [6] PANG, G., KWAN, T., CHAN, C.-H., et al. “LED traffic light as a communications device”. In: *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pp. 788–793, Tokio, Japan, Oct 1999.
- [7] ARNON, S. *Visible Light Communication*. Cambridge, UK, Cambridge University Press, 2015.
- [8] SCHMID, S., RICHNER, T., MANGOLD, S., et al. “EnLighting: an indoor visible light communication system based on networked light bulbs”. In: *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, London, UK, 2016.
- [9] LI, L., HU, P., PENG, C., et al. “Epsilon: a visible light based positioning system”. In: *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, pp. 331–343, Seattle, WA, Apr 2014.

- [10] ZHUANG, Y., HUA, L., QI, L., et al. “A survey of positioning systems using visible LED lights”, *IEEE Communications Surveys Tutorials*, v. PP, n. 99, pp. 1–1, 2018.
- [11] “Cisco visual networking index (VNI) global mobile data traffic forecast update”. . Available at: <<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>>. Online, accessed 02-17-2018.
- [12] ZHAO, S., XU, J., TRESCASES, O. “A dimmable LED driver for visible light communication (VLC) based on LLC resonant DC-DC converter operating in burst mode”. In: *Twenty-Eighth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pp. 2144–2150, Long Beach, USA, Mar 2013.
- [13] U.S. DEPARTMENT OF ENERGY SOLID-STATE LIGHTING PROGRAM. “Energy savings forecast of solid-State lighting in general applications september 2016”. . Available at: <[https://energy.gov/sites/prod/files/2016/09/f33/energysavingsforecast16\\_2.pdf](https://energy.gov/sites/prod/files/2016/09/f33/energysavingsforecast16_2.pdf)>. Online, accessed 02-17-2018.
- [14] JOVICIC, A., LI, J., RICHARDSON, T. “Visible light communication: opportunities, challenges and the path to market”, *IEEE Communications Magazine*, v. 51, n. 12, pp. 26–32, Dec 2013.
- [15] U.S. DEPARTMENT OF ENERGY SOLID-STATE LIGHTING PROGRAM, . Available at: <<https://www.energy.gov/eere/ssl/led-basics>>. Online, accessed 02-17-2018.
- [16] KAVEHRAD, M. “Sustainable energy-efficient wireless applications using light”, *IEEE Communications Magazine*, v. 48, n. 12, pp. 66–73, Dec 2010.
- [17] KAHN, J. M., BARRY, J. R. “Wireless infrared communications”, *Proceedings of the IEEE*, v. 85, n. 2, pp. 265–298, Feb 1997.
- [18] ARMSTRONG, J., SCHMIDT, B. J. C. “Comparison of asymmetrically clipped optical OFDM and DC-biased optical OFDM in AWGN”, *IEEE Communications Letters*, v. 12, n. 5, pp. 343–345, May 2008.
- [19] DIMITROV, S., SINANOVIC, S., HAAS, H. “Clipping noise in OFDM-based optical wireless communication systems”, *IEEE Transactions on Communications*, v. 60, n. 4, pp. 1072–1081, Apr 2012.

- [20] DISSANAYAKE, S. D., ARMSTRONG, J. “Comparison of ACO-OFDM, DCO-OFDM and ADO-OFDM in IM/DD systems”, *Journal of Lightwave Technology*, v. 31, n. 7, pp. 1063–1072, Apr 2013.
- [21] ARMSTRONG, J., SCHMIDT, B. J. C., KALRA, D., et al. “Performance of asymmetrically clipped optical OFDM in AWGN for an intensity modulated direct detection system”. In: *IEEE Globecom*, pp. 1–5, San Francisco, USA, Nov 2006.
- [22] LEE, S. C. J., RANDEL, S., BREYER, F., et al. “PAM-DMT for intensity-modulated and direct-detection optical communication systems”, *IEEE Photonics Technology Letters*, v. 21, n. 23, pp. 1749–1751, Dec 2009.
- [23] DISSANAYAKE, S. D., PANTA, K., ARMSTRONG, J. “A novel technique to simultaneously transmit ACO-OFDM and DCO-OFDM in IM/DD systems”. In: *IEEE GLOBECOM Workshops*, pp. 782–786, Houston, USA, Dec 2011.
- [24] BURNS, R., OF ELECTRICAL ENGINEERS, I. *Communications: An International History of the Formative Years*. London, UK, Institution of Engineering and Technology, 2004.
- [25] BELL, A. G., ADAMS, W. G., TYNDALL, et al. “Discussion on "The photophone and the conversion of radiant energy into sound"”, *Telegraph Engineers, Journal of the Society of*, v. 9, n. 34, pp. 375–383, Jan 1880.
- [26] TANAKA, Y., KOMINE, T., HARUYAMA, S., et al. “Indoor visible communication utilizing plural white LEDs as lighting”. In: *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, v. 2, pp. 81–85, San Diego, CA, USA, Sept 2001.
- [27] KOMINE, T., NAKAGAWA, M. “Integrated system of white LED visible-light communication and power-line communication”, *IEEE Transactions on Consumer Electronics*, v. 49, n. 1, pp. 71–79, Feb 2003.
- [28] TANAKA, Y., KOMINE, T., HARUYAMA, S., et al. “Indoor visible light data transmission system utilizing white LED lights”, *IEICE Transactions on Communications*, v. E86-B, n. 8, pp. 2440–2454, Aug 2003.
- [29] GONZALEZ, O., PEREZ-JIMENEZ, R., RODRIGUEZ, S., et al. “OFDM over indoor wireless optical channel”, *IEE Proceedings - Optoelectronics*, v. 152, n. 4, pp. 199–204, Aug 2005.

- [30] AFGANI, M. Z., HAAS, H., ELGALA, H., et al. “Visible light communication using OFDM”. In: *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, pp. 129–134, Barcelona, Spain, Mar 2006.
- [31] VUCIC, J., KOTTKE, C., NERRETER, S., et al. “White light wireless transmission at 200+ Mb/s net data rate by use of discrete-multitone modulation”, *IEEE Photonics Technology Letters*, v. 21, n. 20, pp. 1511–1513, Oct 2009.
- [32] VUCIC, J., KOTTKE, C., NERRETER, S., et al. “513 Mbit/s visible light communications link based on DMT-modulation of a white LED”, *Journal of Lightwave Technology*, v. 28, n. 24, pp. 3512–3518, Dec 2010.
- [33] KHALID, A. M., COSSU, G., CORSINI, R., et al. “1-Gb/s transmission over a phosphorescent white LED by using rate-adaptive discrete multitone modulation”, *IEEE Photonics Journal*, v. 4, n. 5, pp. 1465–1473, Oct 2012.
- [34] COSSU, G., KHALID, A. M., CHOUDHURY, P., et al. “3.4 Gbit/s visible optical wireless transmission based on RGB LED”, *Opt. Express*, v. 20, n. 26, pp. B501–B506, Dec 2012.
- [35] HUANG, X., SHI, J., LI, J., et al. “750Mbit/s visible light communications employing 64 QAM-OFDM based on amplitude equalization circuit”. In: *Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, Mar 2015.
- [36] WANG, Y., TAO, L., HUANG, X., et al. “Enhanced performance of a high-speed WDM CAP64 VLC system employing volterra series-based nonlinear equalizer”, *IEEE Photonics Journal*, v. 7, n. 3, pp. 1–7, Jun 2015.
- [37] HUSSEIN, A. T., ALRESHEEDI, M. T., ELMIRGHANI, J. M. H. “25 Gbps mobile visible light communication system employing fast adaptation techniques”. In: *18th International Conference on Transparent Optical Networks (ICTON)*, pp. 1–7, 2016.
- [38] BOROGOVCAC, T., RAHAIM, M. B., TUGANBAYEVA, M., et al. “Lights-off visible light communications”. In: *IEEE GLOBECOM Workshops*, pp. 797–801, Houston, USA, Dec 2011.
- [39] YOU, X., CHEN, J., YU, C., et al. “Efficient transmission under low dimming control levels in indoor visible light communications”. In: *10th In-*

*ternational Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 1–5, Budapest, Hungary, Jul 2016.

- [40] LV, X., LI, J. “New miller codes for run-length control in visible light communications”, *IEEE Transactions on Wireless Communications*, v. PP, n. 99, pp. 1–1, 2017.
- [41] YING, K., QIAN, H., BAXLEY, R. J., et al. “Joint optimization of precoder and equalizer in MIMO VLC systems”, *IEEE Journal on Selected Areas in Communications*, v. 33, n. 9, pp. 1949–1958, Sept 2015.
- [42] MOSTAFA, A., LAMPE, L. “Optimal and robust beamforming for secure transmission in MISO visible-light communication links”, *IEEE Transactions on Signal Processing*, v. 64, n. 24, pp. 6501–6516, Dec 2016.
- [43] STEPNIAK, G., SIUZDAK, J., ZWIERKO, P. “Compensation of a VLC phosphorescent white LED nonlinearity by means of Volterra DFE”, *IEEE Photonics Technology Letters*, v. 25, n. 16, pp. 1597–1600, Aug 2013.
- [44] LI, X., CHEN, H., LI, S., et al. “Volterra-based nonlinear equalization for nonlinearity mitigation in organic VLC”. In: *13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 616–621, Valencia, Spain, Jun 2017.
- [45] ELGALA, H., MESLEH, R., HAAS, H. “Predistortion in optical wireless transmission using OFDM”. In: *Ninth International Conference on Hybrid Intelligent Systems*, v. 2, pp. 184–189, Shenyang, China, Aug 2009.
- [46] ÁVILA, F. R., LOVISOLO, L. “Unsupervised time domain nonlinear post-equalization for ACO-OFDM visible light communication systems”. In: *25th European Signal Processing Conference (EUSIPCO)*, pp. 848–852, Kos Island, Greece, Aug 2017.
- [47] MITRA, R., BHATIA, V. “Low complexity post-distorter for visible light communications”, *IEEE Communications Letters*, v. 21, n. 9, pp. 1977–1980, Sept 2017.
- [48] LAZZARIN, G., PUPOLIN, S., SARTI, A. “Nonlinearity compensation in digital radio systems”, *IEEE Transactions on Communications*, v. 42, n. 234, pp. 988–999, Feb 1994.
- [49] BENEDETTO, S., BIGLIERI, E., DAFFARA, R. “Modeling and Performance Evaluation of Nonlinear Satellite Links—A Volterra Series Approach”,

*IEEE Transactions on Aerospace and Electronic Systems*, v. 15, n. 4, pp. 494–507, Jul 1979.

- [50] HE, W., CHAN, Z., SUN, C., et al. “Adaptive neural network control of a flapping wing micro aerial vehicle with disturbance observer”, *IEEE Transactions on Cybernetics*, v. 47, n. 10, pp. 3452–3465, Oct 2017.
- [51] HE, W., DONG, Y., SUN, C. “Adaptive neural impedance control of a robotic manipulator with input saturation”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 46, n. 3, pp. 334–344, Mar 2016.
- [52] HE, W., GE, S. S. “Cooperative control of a nonuniform gantry crane with constrained tension”, *Automatica*, v. 66, n. 4, pp. 146–154, Jan 2016.
- [53] HE, W., CHEN, Y., YIN, Z. “Adaptive Neural Network Control of an Uncertain Robot with Full-State Constraints”, *IEEE Transactions on Cybernetics*, v. 46, n. 3, pp. 620–629, Mar 2016.
- [54] GUERIN, A., FAUCON, G., BOUQUIN-JEANNES, R. L. “Nonlinear acoustic echo cancellation based on Volterra filters”, *IEEE Transactions on Speech and Audio Processing*, v. 11, n. 6, pp. 672–683, Nov 2003.
- [55] AZPICUETA-RUIZ, L. A., ZELLER, M., FIGUEIRAS-VIDAL, A. R., et al. “Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 19, n. 1, pp. 97–110, Jan 2011.
- [56] LIU, W., PRINCIPE, J., HAYKIN, S. *Kernel Adaptive Filtering: A Comprehensive Introduction*. Hoboken, NJ, USA, Wiley, 2011.
- [57] TAKIZAWA, M. A., YUKAWA, M. “Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel Hilbert space”, *IEEE Transactions on Signal Processing*, v. 63, n. 16, pp. 4257–4269, Aug 2015.
- [58] LI, K., PRÍNCIPE, J. C. “The Kernel Adaptive Autoregressive-Moving-Average Algorithm”, *IEEE Transactions on Neural Networks and Learning Systems*, v. 27, n. 2, pp. 334–346, Feb 2016.
- [59] MATHEWS, V. J. “Adaptive polynomial filters”, *IEEE Signal Processing Magazine*, v. 8, n. 3, pp. 10–26, Jul 1991.
- [60] BATISTA, E. L. O., TOBIAS, O. J., SEARA, R. “A fully LMS adaptive interpolated Volterra structure”. In: *IEEE International Conference on*

*Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3613–3616, Las Vegas, USA, Mar 2008.

- [61] LUIZ, E., BATISTA, O., SEARA, R. “A novel reduced-rank approach for implementing Volterra filters”. In: *24th European Signal Processing Conference (EUSIPCO)*, pp. 1778–1782, Budapest, Hungary, Aug 2016.
- [62] BIGLIERI, E., GERSHO, A., GITLIN, R., et al. “Adaptive Cancellation of Non-linear Intersymbol Interference for Voiceband Data Transmission”, *IEEE Journal on Selected Areas in Communications*, v. 2, n. 5, pp. 765–777, Sept 1984.
- [63] ZELLER, M., KELLERMANN, W. “Fast and robust adaptation of DFT-domain Volterra filters in diagonal coordinates using iterated coefficient updates”, *IEEE Transactions on Signal Processing*, v. 58, n. 3, pp. 1589–1604, Mar 2010.
- [64] BATISTA, E. L. O., TOBIAS, O. J., SEARA, R. “A sparse-interpolated scheme for implementing adaptive Volterra filters”, *IEEE Transactions on Signal Processing*, v. 58, n. 4, pp. 2022–2035, Apr 2010.
- [65] ENZINGER, H., FREIBERGER, K., KUBIN, G., et al. “Fast time-domain Volterra Filtering”. In: *Asilomar Conference on Signals, Systems and Computers*, pp. 225–228, Pacific Grove, CA, USA, Nov 2016.
- [66] PARK, J., CHANG, J. H. “Frequency-domain Volterra filter based on data-driven soft decision for nonlinear acoustic echo suppression”, *IEEE Signal Processing Letters*, v. 21, n. 9, pp. 1088–1092, Sept 2014.
- [67] CRISTALDI, L., FAIFER, M., LAURANO, C., et al. “Iterative method for the definition of frequency-domain Volterra models”. In: *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, Torino, Italy, May 2017.
- [68] DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation*. 4 ed. New York, NY, USA, Springer, 2012.
- [69] YAZDANPANA, H., DINIZ, P. S. R., LIMA, M. V. S. “A simple set-membership affine projection algorithm for sparse system modeling”. In: *24th European Signal Processing Conference (EUSIPCO)*, pp. 1798–1802, Budapest, Hungary, Aug 2016.
- [70] WERNER, S., APOLINÁRIO, J. A., DINIZ, P. S. R., et al. “A set-membership approach to normalized proportionate adaptation algorithms”. In: *2005*

*13th European Signal Processing Conference*, pp. 1–4, Antalya, Turkey, Sept 2005.

- [71] WERNER, S., APOLINÁRIO, J. A., DINIZ, P. S. R. “Set-membership proportionate affine projection algorithms”, *EURASIP Journal on Audio, Speech, and Music Processing*, v. 2007, n. 1, pp. 10–10, 2007.
- [72] S. NAGARAJ, S. GOLLAMUDI, S. K., HUANG, Y. F. “BEACON: an adaptive set-membership filtering technique with sparse updates”, *IEEE Transactions on Signal Processing*, v. 47, n. 11, pp. 2928–2941, Nov 1999.
- [73] ARABLOUEI, R., DOGANÇAY, K. “Set-membership recursive least-squares adaptive filtering algorithm”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3765–3768, Mar 2012.
- [74] ARENAS-GARCIA, J., AZPICUETA-RUIZ, L. A., SILVA, M. T. M., et al. “Combinations of adaptive filters: performance and convergence properties”, *IEEE Signal Processing Magazine*, v. 33, n. 1, pp. 120–140, Jan 2016.
- [75] CHAMON, L. F. O., LOPES, W. B., LOPES, C. G. “Combination of adaptive filters with coefficients feedback”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3785–3788, Kyoto, Japan, 2012.
- [76] MARTINS, W. A., LIMA, M. V. S., DINIZ, P. S. R. “Semi-blind data-selective equalizers for QAM”. In: *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, pp. 501–505, Recife, Brazil, Jul 2008.
- [77] WANG, Q., GIUSTINIANO, D., PUCCINELLI, D. “OpenVLC: software-defined open architecture for embedded visible light networks”. In: *1st ACM Workshop on Visible Light Communication Systems*, pp. 15–20, Maui, HI, USA, Sept 2014.
- [78] SEDRA, A. S., SMITH, K. C. *Microelectronic Circuits*. 1 ed. New York, NY, USA, 1998.
- [79] SCHUBERT, E. F. *Light-Emitting Diodes*. London, UK, Cambridge University Press, 2006.
- [80] STEIGERWALD, D. A., BHAT, J. C., COLLINS, D., et al. “Illumination with solid state lighting technology”, *IEEE Journal of Selected Topics in Quantum Electronics*, v. 8, n. 2, pp. 310–320, Mar 2002.

- [81] GRUBOR, J., RANDEL, S., LANGER, K. D., et al. “Broadband information broadcasting using LED-based interior lighting”, *Journal of Lightwave Technology*, v. 26, n. 24, pp. 3883–3892, Dec 2008.
- [82] ARMSTRONG, J., GREEN, R. J., HIGGINS, M. D. “Comparison of three receiver designs for optical wireless communications using white LEDs”, *IEEE Communications Letters*, v. 16, n. 5, pp. 748–751, May 2012.
- [83] GFELLER, F. R., BAPST, U. “Wireless in-house data communication via diffuse infrared radiation”, *Proceedings of the IEEE*, v. 67, n. 11, pp. 1474–1486, Nov 1979.
- [84] CUEVAS, A. “The recombination parameter  $J_0$ ”. In: *Proceedings of the 4th International Conference on Crystalline Silicon Photovoltaics*, v. 55, pp. 53 – 62, s-Hertogenbosch, Netherlands, 2014.
- [85] MCCULLAGH, M. J., WISELY, D. R. “155 Mbit/s optical wireless link using a bootstrapped silicon APD receiver”, *Electronics Letters*, v. 30, n. 5, pp. 430–432, Mar 1994.
- [86] DIMITROV, S., HAAS, H. *Principles of LED Light Communications: Towards Networked Li-Fi*. London, UK, Cambridge University Press, 2015.
- [87] VUCIC, J., KOTTKE, C., HABEL, K., et al. “803 Mbit/s visible light WDM link based on DMT modulation of a single RGB LED luminary”. In: *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*, pp. 1–3, Los Angeles, USA, Mar 2011.
- [88] LEE, E. A., MESSERSCHMITT, D. G. *Digital Communication*. AZ Dordrecht, Netherlands, Springer, 2012.
- [89] WIDROW, B., MCCOOL, J. M., LARIMORE, M. G., et al. “Stationary and nonstationary learning characteristics of the LMS adaptive filter”, *Proceedings of the IEEE*, v. 64, n. 8, pp. 1151–1162, Aug 1976.
- [90] OZEKI, K., UMEDA, T. “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties”, *Electronics and Communications in Japan (Part I: Communications)*, v. 67, n. 5, pp. 19–27, 1984.
- [91] HAYKIN, S. S. *Adaptive Filter Theory*. 4 ed. Carmel, IN, USA, Pearson Education, 2008.

- [92] RAO, A. K., HUANG, Y. F. “Tracking characteristics of an OBE parameter-estimation algorithm”, *IEEE Transactions on Signal Processing*, v. 41, n. 3, pp. 1140–1148, Mar 1993.
- [93] GOLLAMUDI, S., KAPOOR, S., NAGARAJ, S., et al. “Set-membership adaptive equalization and an updatior-shared implementation for multiple channel communications systems”, *IEEE Transactions on Signal Processing*, v. 46, n. 9, pp. 2372–2385, Sept 1998.
- [94] DE LAMARE, R. C., DINIZ, P. S. R. “Set-membership adaptive algorithms based on time-varying error bounds for CDMA interference suppression”, *IEEE Transactions on Vehicular Technology*, v. 58, n. 2, pp. 644–654, Feb 2009.
- [95] GOODWIN, G. C., PAYNE, R. L. *Dynamic System Identification: Experiment Design and Data Analysis*. Cambridge, MA, USA, Academic Press, 1977.
- [96] GIANNAKIS, G. B., SERPEDIN, E. “Linear multichannel blind equalizers of nonlinear FIR Volterra channels”, *IEEE Transactions on Signal Processing*, v. 45, n. 1, pp. 67–81, Jan 1997.
- [97] MESLEH, R., ELGALA, H., HAAS, H. “LED nonlinearity mitigation techniques in optical wireless OFDM communication systems”, *IEEE/OSA Journal of Optical Communications and Networking*, v. 4, n. 11, pp. 865–875, Nov 2012.
- [98] DA SILVA, F. B., MARTINS, W. A. “A computational platform for visible light communications”. In: *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2017)*, pp. 891–895, São Pedro, Brazil, Sept 2017.
- [99] KUECH, F., KELLERMANN, W. “Proportionate NLMS algorithm for second-order Volterra filters and its application to nonlinear echo cancellation”. In: *Int. Workshop Acoust. Echo Noise Control (IWAENC)*, pp. 75–78, Kyoto, Japan, 2003.
- [100] LIMA, M. V. S., FERREIRA, T. N., MARTINS, W. A., et al. “Sparsity-aware data-selective adaptive filters”, *IEEE Transactions on Signal Processing*, v. 62, n. 17, pp. 4557–4572, Sept 2014.
- [101] DOUGLAS, S. C. “Adaptive filters employing partial updates”, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 44, n. 3, pp. 209–216, Mar 1997.

- [102] MARTINEZ-RAMON, M., ARENAS-GARCIA, J., NAVIA-VAZQUEZ, A., et al. “An adaptive combination of adaptive filters for plant identification”. In: *14th International Conference on Digital Signal Processing Proceedings. DSP 2002*, v. 2, pp. 1195–1198, Santorini, Greece, Jul 2002.
- [103] NASCIMENTO, V. H., DE LAMARE, R. C. “A low-complexity strategy for speeding up the convergence of convex combinations of adaptive filters”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3553–3556, Mar 2012.
- [104] . Available at: [http://www1.futureelectronics.com/doc/EVERLIGHT%C2%A0/334-15\\_\\_T1C1-4WYA.pdf](http://www1.futureelectronics.com/doc/EVERLIGHT%C2%A0/334-15__T1C1-4WYA.pdf). Online, accessed 02-17-2018.
- [105] BIAGI, M., BOROGOVAC, T., LITTLE, T. D. C. “Adaptive receiver for indoor visible light communications”, *Journal of Lightwave Technology*, v. 31, n. 23, pp. 3676–3686, Dec 2013.
- [106] ELGALA, H., MESLEH, R., HAAS, H. “An LED model for intensity-modulated optical communication systems”, *IEEE Photonics Technology Letters*, v. 22, n. 11, pp. 835–837, Jun 2010.
- [107] MORGAN, D. R., MA, Z., KIM, J., et al. “A generalized memory polynomial model for digital predistortion of RF power amplifiers”, *IEEE Transactions on Signal Processing*, v. 54, n. 10, pp. 3852–3860, Oct 2006.
- [108] KIBANGOU, A. Y., FAVIER, G. “Tensor analysis-based model structure determination and parameter estimation for block-oriented nonlinear systems”, *IEEE Journal of Selected Topics in Signal Processing*, v. 4, n. 3, pp. 514–525, Jun 2010.
- [109] ZHOU, D., DEBRUNNER, V. “Efficient adaptive nonlinear filters for nonlinear active noise control”, *IEEE Transactions on Circuits and Systems*, v. 54, n. 3, pp. 669–681, Mar 2007.
- [110] BERENQUER, P. W., NÖLLE, M., MOLLE, L., et al. “Nonlinear digital predistortion of transmitter components”, *Journal of Lightwave Technology*, v. 34, n. 8, pp. 1739–1745, Apr 2016.
- [111] AGUIRRE, L. A., COELHO, M. C. S., CORREA, M. V. “On the interpretation and practice of dynamical differences between Hammerstein and Wiener models”, *IEE Proceedings - Control Theory and Applications*, v. 152, n. 4, pp. 349–356, Jul 2005.

- [112] CHEN, H. F. “Adaptive regulator for Hammerstein and Wiener systems with noisy observations”, *IEEE Transactions on Automatic Control*, v. 52, n. 4, pp. 703–709, Apr 2007.
- [113] BARRY, J. R., KAHN, J. M., KRAUSE, W. J., et al. “Simulation of multipath impulse response for indoor wireless optical channels”, *IEEE Journal on Selected Areas in Communications*, v. 11, n. 3, pp. 367–379, Apr 1993.
- [114] LEE, K., PARK, H., BARRY, J. R. “Indoor channel characteristics for visible light communications”, *IEEE Communications Letters*, v. 15, n. 2, pp. 217–219, Feb 2011.
- [115] CARRUTHERS, J. B., KAHN, J. M. “Modeling of nondirected wireless infrared channels”, *IEEE Transactions on Communications*, v. 45, n. 10, pp. 1260–1268, Oct 1997.
- [116] DA SILVA, F. B., Mar 2018. Available at: <[https://github.com/fbs2112/adaptive\\_multifilters](https://github.com/fbs2112/adaptive_multifilters)>. Online, accessed 02-20-2018. Public access.