

REDUCED COMPLEXITY ADAPTIVE FILTERING ALGORITHMS WITH APPLICATIONS TO COMMUNICATIONS SYSTEMS

Stefan Werner

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Auditorium S3 at Helsinki University of Technology (Espoo, Finland) on the 15th of November, 2002, at 12 o'clock noon.

**Helsinki University of Technology
Department of Electrical and Communications Engineering
Signal Processing Laboratory**

**Teknillinen korkeakoulu
Sähkö- ja tietoliikennetekniikan osasto
Signaalinkäsittelytekniikan laboratorio**

Distribution:
Helsinki University of Technology
Signal Processing Laboratory
P.O. Box 3000
FIN-02015 HUT
Tel. +358-9-451 3211
Fax. +358-9-452 3614
E-mail: Mirja.Lemetyinen@hut.fi

© Stefan Werner

ISBN 951-22-6175-8
ISSN 1458-6401

Otamedia Oy
Espoo 2002

Abstract

This thesis develops new adaptive filtering algorithms suitable for communications applications with the aim of reducing the computational complexity of the implementation. Low computational complexity of the adaptive filtering algorithm can, for example, reduce the required power consumption of the implementation. A low power consumption is important in wireless applications, particularly at the mobile terminal side, where the physical size of the mobile terminal and long battery life are crucial. We focus on the implementation of two types of adaptive filters: linearly-constrained minimum-variance (LCMV) adaptive filters and conventional training-based adaptive filters.

For LCMV adaptive filters, normalized data-reusing algorithms are proposed which can trade off convergence speed and computational complexity by varying the number of data-reuses in the coefficient update. Furthermore, we propose a transformation of the input signal to the LCMV adaptive filter, which properly reduces the dimension of the coefficient update. It is shown that transforming the input signal using successive Householder transformations renders a particularly efficient implementation. The approach allows any unconstrained adaptation algorithm to be applied to linearly constrained problems.

In addition, a family of algorithms is proposed using the framework of set-membership filtering (SMF). These algorithms combine a bounded error specification on the adaptive filter with the concept of data-reusing. The resulting algorithms have low average computational complexity because coefficient update is not performed at each iteration. In addition, the adaptation algorithm can be adjusted to achieve a desired computational complexity by allowing a variable number of data-reuses for the filter update.

Finally, we propose a framework combining sparse update in time with sparse update of filter coefficients. This type of partial-update (PU) adaptive filters are suitable for applications where the required order of the adaptive filter is conflicting with tight constraints for the processing power.

Acknowledgements

This work was financially supported by the Graduate School in Electronics Telecommunications and Automation (GETA), the Finnish Academy, National Technology Agency (TEKES), and Nokia foundation.

I would like to begin by expressing my sincere thanks to Prof. Timo Laakso who has been my supervisor during the course of my work. Timo has been a great support during my time at the university. His constructive criticism has contributed to the technical quality of the manuscript. During the years, Timo has probably been correcting hundreds of my drafts for technical documents. My skills for writing technical documents has gone from poor to what it is today. Already, at the very beginning of my studies, Timo taught me how to write and present research in an objective manner. For this, I am truly grateful.

I wish to express my appreciation to my co-supervisor, co-author and friend Prof. Marcello de Campos. His constructive criticism and comments has contributed to the technical content and visual form of my thesis. During the years we have developed a very efficient and strong cooperation. I just wonder why many of our original ideas appear when having a beer in front of us? Fortunately, many of these ideas led to results presented in this thesis. My many research visits to Brazil have been possible due to the hospitality of Marcello (and family). Thanks!

I truly appreciate the support, supervision and friendship of Prof. Paulo Diniz who is co-author in many scientific papers. Many of the results from our fruitful cooperation during the years can be found in this thesis. Paulo has been like a third supervisor, and I have learned a lot from his experience as a researcher (and also a great deal of non-printable jokes!). I hope and believe our co-operation continues in the future.

I would also like to thank my co-author and friend Prof. José Apolinário. Every time I visit Rio, a visit to the Military Institute is on the agenda.

I would like to thank my thesis pre-examiners Prof. Yih-Fang Huang and Prof. Markku Juntti for their constructive comments. I am sure their comments improved the final manuscript.

I am grateful to Prof. Iiro Hartimo, the director of GETA, to GETA secretary Marja

Leppäharju, and to our laboratory secretaries Anne Jääskeläinen and Mirja Lemetyinen for the help with all the practical issues and arrangements. Prof. Visa Koivunen deserves thanks for giving valuable hints in the final stage of the manuscript preparation.

My impression of Finns in general before actually moving to Finland was that they were shy and did not talk more than absolutely necessary. Of course this was completely wrong and my good friend and colleague Tomi is a living proof that Finns can be as noisy and cheerful as in any other place on earth. My friend Pasi has saved me from eating at the university restaurant on a daily basis. I would like to thank my colleagues with whom I had many interesting discussions, in particular, Yaohui, Martin, and Mihai.

During my studies I was fortunate to spend one year in the Signal Processing Laboratory at the Federal University of Rio de Janeiro, Brazil. I would like to direct a special thank to the people in the laboratory for creating such a friendly and relaxed environment (*muito obrigado!*). I would like to thank friends in Rio who helped me in any kind of daily-life emergency (beer or bureaucracy): my partner-in-crime 'A.-M.' with family, 'Prof. F.' with family, 'Batman' with family, Ailton ('Cabeça Suja'), Cássio ('Cabeção'), and Heveline. Unfortunately, most of them preferred to spend their time in bars instead of working but I kindly accepted their invitations whenever I had time (*e eu sempre tinha tempo!*).

The deepest gratitude goes to my family for their love and support: my parents Lennart and Yvonne, my brothers Mikael and Thomas, sister-in-laws Helena and Anne, and my wonderful nieces and nephews Sofie, Fredrik, Anna, Samuel and Emilia. In the end, it is my family who taught me the valuable things in life. All my good behavior comes from them (the bad behavior I have adopted from my friends.) My second family Seppo and Eija, Anne, and Johanna deserves my sincere thanks for welcoming me into their home with open arms, although being Swedish.

Finally, the work put into this thesis was worthwhile due to Katja and Felix – to whom I dedicate this thesis.

Helsinki, October 15, 2002

Stefan Werner

Contents

| | |
|--|------------|
| Abstract | i |
| Acknowledgements | iii |
| Abbreviations and Acronymns | xi |
| List of Symbols | xv |
| List of Tables | xix |
| List of Figures | xxi |
| 1 Introduction | 1 |
| 1.1 Motivation | 1 |
| 1.2 Adaptive Filters | 2 |
| 1.3 Applications | 4 |
| 1.3.1 System Identification | 4 |
| 1.3.2 Interference Suppression in Communications Systems | 5 |
| 1.3.3 Antenna Arrays | 6 |
| 1.4 Adaptation Algorithms | 7 |
| 1.4.1 The Least Mean-Square (LMS) Algorithm | 9 |
| 1.4.2 The Recursive Least-Squares (RLS) Algorithm | 11 |
| 1.4.3 The Affine-Projection (AP) Algorithm | 12 |
| 1.4.4 The Quasi-Newton (QN) Algorithm | 14 |
| 1.4.5 Complex-Valued Signals | 15 |

| | | |
|----------|--|-----------|
| 1.5 | Overview and Contributions | 16 |
| 2 | Constrained Adaptive Filters | 21 |
| 2.1 | Introduction | 21 |
| 2.2 | Optimal Linearly-Constrained Minimum-Variance Filter | 24 |
| 2.3 | LC Adaptive Filtering | 25 |
| 2.3.1 | The Constrained Least Mean-Square Algorithm | 25 |
| 2.3.2 | The Constrained Least-Squares Algorithm | 27 |
| 2.4 | The Generalized Sidelobe Canceling Model | 30 |
| 2.4.1 | The GSC-RLS Algorithm | 33 |
| 2.5 | Equivalence of LC and GSC Adaptive Filters | 33 |
| 2.6 | Choice of Blocking Matrix \mathbf{B} | 36 |
| 2.7 | Simulation Results | 38 |
| 2.7.1 | Beamforming with Derivative Constraints | 38 |
| 2.7.2 | Identification of Plant with Linear Phase | 39 |
| 2.8 | Conclusions | 41 |
| | Appendix A2.1 | 43 |
| 3 | Householder Constrained Adaptation Algorithms | 47 |
| 3.1 | Introduction | 47 |
| 3.2 | Householder-Transform Constrained Algorithms | 49 |
| 3.2.1 | Choice of the Transformation Matrix \mathbf{Q} | 50 |
| 3.2.2 | The Normalized HCLMS Algorithm | 52 |
| 3.2.3 | Computational Complexity Issues | 52 |
| 3.2.4 | Householder-Transform Constrained Algorithms and the GSC | 55 |
| 3.3 | Simulation Results | 56 |
| 3.4 | Conclusions | 57 |
| 4 | Set-Membership Binormalized Data Reusing LMS Algorithms | 63 |
| 4.1 | Introduction | 63 |
| 4.2 | Set-Membership Filtering | 65 |

| | | |
|----------|---|------------|
| 4.2.1 | Set-Membership Normalized LMS Algorithm | 67 |
| 4.2.2 | SM-NLMS Algorithm – Convergence Issues | 68 |
| 4.3 | Set-Membership Binormalized Data-Reusing LMS Algorithms | 70 |
| 4.3.1 | Algorithm I | 71 |
| 4.3.2 | Algorithm II | 75 |
| 4.3.3 | Computational Complexity | 78 |
| 4.4 | Second-Order Statistical Analysis | 80 |
| 4.4.1 | Coefficient-Error Vector | 80 |
| 4.4.2 | Input-Signal Model | 81 |
| 4.4.3 | Excess MSE for White Input Signals | 82 |
| 4.4.4 | Excess MSE for Colored Input Signals | 84 |
| 4.5 | Simulation Results | 85 |
| 4.6 | Conclusions | 92 |
| | Appendix A4.1 | 92 |
| | Appendix A4.2 | 96 |
| | Appendix A4.3 | 97 |
| 5 | Set-Membership Affine-Projection Algorithm | 99 |
| 5.1 | Introduction | 99 |
| 5.2 | Set-Membership Affine-Projection Algorithm | 100 |
| 5.2.1 | Derivation of the General SM-AP Algorithm | 101 |
| 5.2.2 | The Parameter Vector $\mathbf{g}(k)$ | 103 |
| 5.3 | Simulation Results | 105 |
| 5.3.1 | System Identification | 105 |
| 5.3.2 | Adaptive Equalizer | 106 |
| 5.4 | Conclusions | 108 |
| 6 | Low-Complexity Constrained Affine-Projection Algorithms | 111 |
| 6.1 | Introduction | 111 |
| 6.2 | The Constrained Affine-Projection Algorithm | 113 |

| | | |
|----------|---|------------|
| 6.3 | The Householder-Transform CAP Algorithm | 114 |
| 6.4 | Set-Membership Constrained Affine-Projection Algorithm | 116 |
| 6.5 | On the Convergence of the CAP Algorithms | 119 |
| 6.6 | Simulation Results | 123 |
| 6.6.1 | Identification of Plant with Linear Phase | 123 |
| 6.6.2 | Linearly-Constrained Minimum-Variance Filtering of Sinusoids | 125 |
| 6.6.3 | Interference Suppression in a CDMA Communications System | 127 |
| 6.7 | Conclusions | 130 |
| 7 | Partial-Update NLMS Algorithms with Data-Selective Updating | 133 |
| 7.1 | Introduction | 134 |
| 7.2 | The Partial-Update NLMS Algorithm | 136 |
| 7.3 | The Set-Membership Partial-Update NLMS Algorithm | 138 |
| 7.3.1 | Algorithm Derivation | 139 |
| 7.3.2 | Convergence Issues | 143 |
| 7.4 | Computational Complexity | 147 |
| 7.5 | Simulation Results | 148 |
| 7.5.1 | Verification of the Analysis of the PU-NLMS Algorithm | 148 |
| 7.5.2 | SM-PU-NLMS Algorithm | 149 |
| 7.6 | Conclusions | 152 |
| | Appendix A7.1 | 154 |
| | Appendix A7.2 | 155 |
| | Appendix A7.3 | 159 |
| 8 | Conclusions and Future Work | 165 |
| 8.1 | Conclusions | 165 |

| | |
|---------------------------|------------|
| 8.2 Future Work | 168 |
| Bibliography | 170 |

List of Abbreviations and Acronyms

| | |
|----------|--|
| AP | affine-projection |
| APOC | <i>a posteriori</i> output constrained |
| APOC-LMS | <i>a posteriori</i> output constrained LMS |
| BEACON | bounding ellipsoidal adaptive constrained |
| BNDRCLMS | binormalized data-reusing constrained LMS |
| BNDRLMS | binormalized data-reusing LMS |
| BUC | block underdetermined covariance |
| CAP | constrained affine-projection |
| CLMS | constrained LMS |
| CDMA | code division multiple access |
| CORDIC | coordinate rotation digital computer |
| CQN | constrained QN |
| CRLS | constrained RLS |
| DOA | direction of arrival |
| DS-CDMA | direct-sequence CDMA |
| FIR | finite-duration impulse response |
| GSC | generalized sidelobe canceler |
| HC | Householder constrained |
| HCAP | Householder constrained AP |
| HCLMS | Householder constrained LMS |
| HNCLMS | Householder normalized constrained LMS |
| HT | Householder transform |

| | |
|------------|--|
| IIR | infinite-duration impulse response |
| ISI | intersymbol interference |
| LC | linearly constrained |
| LCMV | linearly-constrained minimum-variance |
| LMS | least-mean squares |
| LS | least-squares |
| MOE | mean output energy |
| MSE | mean-squared error |
| NCLMS | normalized constrained LMS |
| NLMS | normalized least mean squares |
| NNDR | normalized new data-reusing |
| NNDRLMS | normalized new data-reusing LMS |
| OBE | optimal bounding ellipsoid |
| OBS | optimal bounding spheroid |
| PA | projection algorithm |
| PU | partial-update |
| QN | quasi-Newton |
| RLS | recursive least-squares |
| SINR | signal-to-interference plus noise ratio |
| SM | set-membership |
| SM-AP | set-membership AP |
| SMART | set-membership adaptive recursion techniques |
| SM-BNDRLMS | set-membership BNDRLMS |
| SMF | set-membership filtering |
| SMI | set-membership identification |
| SM-NLMS | set-membership NLMS |
| SM-PU | set-membership partial-update |
| SNR | signal-to-noise ratio |
| SVD | singular value decomposition |

ULA uniform linear array
WLS weighted least-squares

List of Symbols

| | |
|------------------------|--|
| $x(k)$ | input signal sequence |
| $\mathbf{x}(k)$ | input signal vector |
| $y(k)$ | output signal sequence |
| $d(k)$ | desired signal sequence |
| $e(k)$ | output error sequence |
| $n(k)$ | noise sequence |
| $\mathbf{w}(k)$ | vector of coefficients of the adaptive filter |
| $w_i(k)$ | i th coefficient of the adaptive filter |
| N | number of filter coefficients |
| $J_{\mathbf{w}}$ | objective function |
| $\mathbf{s}(\theta_i)$ | steering vector |
| θ_i | direction of arrival of user i |
| μ | step size |
| $\text{tr}\{\cdot\}$ | trace |
| $E\{x\}$ | expectation of the random variable x |
| \mathbf{R} | autocorrelation matrix |
| $Q(x)$ | Q-function, $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$ |
| \mathbf{p} | cross-correlation vector |
| λ | forgetting factor |
| P | number of data reuse |
| δ | small positive constant |
| $\mathbf{X}(k)$ | input signal matrix |

| | |
|------------------------|---|
| $\mathbf{d}(k)$ | desired signal vector |
| \mathbf{I} | identity matrix |
| M | number of antennas |
| \mathbf{C} | constraint matrix |
| \mathbf{f} | gain vector |
| p | number of constraints |
| \mathbf{F} | fixed vector satisfying the linear constraints |
| \mathbf{P} | projection matrix onto the subspace orthogonal to the subspace spanned by the constraint matrix |
| \mathbf{B} | blocking matrix |
| $\bar{\mathbf{w}}$ | transformed adaptive filter vector |
| $\Delta \mathbf{v}(k)$ | coefficient-vector evolution |
| \mathbb{Z} | set of integer numbers |
| \mathbb{R} | set of real numbers |
| \mathbb{R}^N | set of real N -dimensional vectors |
| $\ \mathbf{x}\ $ | norm of vector \mathbf{x} |
| \perp | perpendicular to |
| \parallel | parallel to |
| \mathbf{Q} | orthogonal transformation matrix |
| $\bar{\mathbf{C}}$ | transformed constraint matrix |
| $\bar{\mathbf{P}}$ | transformed projection matrix |
| γ | bound on magnitude of output error |
| Θ | feasibility set |
| $\mathcal{H}(k)$ | constraint set |
| ψ_k | exact membership set |
| \cap | intersection |
| $\alpha(k)$ | time-varying step size |
| $P_e(k)$ | probability of update |
| $\Delta \mathbf{w}(k)$ | coefficient error vector |

| | |
|---------------------------------|---|
| σ_n^2 | variance of noise sequence |
| σ_x^2 | variance of input sequence |
| $\Delta\xi(k)$ | excess MSE at iteration k |
| ξ_{exc} | excess MSE |
| L | number of coefficients in partial update |
| $\mathcal{I}_L(k)$ | index set of coefficients to be updated |
| $\mathbf{A}_{\mathcal{I}_L(k)}$ | selection matrix for coefficients to be updated |

List of Tables

| | | |
|-----|--|-----|
| 1.1 | The Least Mean-Square Algorithm | 10 |
| 1.2 | The Recursive Least-Squares Algorithm | 11 |
| 1.3 | The Affine-Projection Algorithm | 13 |
| 1.4 | The Quasi-Newton Algorithm | 15 |
| 1.5 | Characterization of the Algorithms Developed in the Thesis. | 18 |
| 2.1 | The Constrained RLS Algorithm. | 31 |
| 2.2 | Signal Parameters | 39 |
| 3.1 | The HCLMS Algorithm | 54 |
| 3.2 | Computation of $\mathbf{Q}\mathbf{x}(k)$ | 54 |
| 3.3 | Construction of Matrix \mathbf{V} Containing the Householder Vectors | 55 |
| 3.4 | Computational Complexity | 56 |
| 3.5 | The HCQN Algorithm | 58 |
| 3.6 | Signal Parameters | 59 |
| 3.7 | Output Gains in the Directions of the Interferers. | 59 |
| 4.1 | The Set-Membership Binormalized LMS Algorithm I | 75 |
| 4.2 | The Set-Membership Binormalized LMS Algorithm II | 79 |
| 4.3 | Computational Complexity per Update | 79 |
| 4.4 | Excess Mean-Square Error in Nonstationary Environments | 91 |
| 5.1 | The Set-Membership Affine-Projection Algorithm. | 106 |
| 6.1 | The Constrained Affine-Projection Algorithm. | 114 |

| | | |
|-----|---|-----|
| 6.2 | The Householder-Transform Constrained Affine-Projection Algorithm . . . | 116 |
| 6.3 | The Set-Membership Constrained Affine-Projection Algorithm. | 119 |
| 7.1 | SM-PU-NLMS Algorithm, L Time-Varying with $L \leq L_{max}$ | 162 |
| 7.2 | SM-PU-NLMS Algorithm, L Fixed During the Adaptation. | 163 |
| 7.3 | Computational Complexity. | 163 |
| 8.1 | Algorithm Comparisons for LCMV Adaptive Filters. | 167 |
| 8.2 | Algorithm Comparisons for Training-Based Adaptive Filters. | 168 |

List of Figures

- 1.1 Schematic diagram of an adaptive filter. 2
- 1.2 Schematic diagram of a system-identification application. 5
- 1.3 Schematic diagram of an adaptive equalizer. 6
- 1.4 Schematic diagram of an LCMV adaptive antenna array consisting of M antennas, with an adaptation algorithm minimizing the objective function $J_{\mathbf{w}}$ while constrained to pass the desired signal from user direction $\mathbf{s}(\theta_1)$ with unity response. 8
- 1.5 Adaptive FIR filter. 9
- 2.1 Broadband adaptive receiving array. 22
- 2.2 Geometrical interpretation of some constrained algorithms. 28
- 2.3 Generalized sidelobe canceling (GSC) model. 32
- 2.4 Coefficient-error vector as a function of the iteration k for a beamforming application using derivative constraints. 40
- 2.5 Norm of the difference of the CRLS and the GSC-RLS coefficient vectors as a function of the iteration k for a beamforming application using derivative constraints. 41
- 2.6 Coefficient-error vector as a function of the iteration k for a system-identification application. 42
- 2.7 Norm of the difference of the CRLS and the GSC-RLS coefficient vectors as a function of the iteration k for a system-identification application. 43
- 3.1 Coefficient-vector rotation. 53

| | | |
|------|--|----|
| 3.2 | The HC adaptive filter under the GSC model. | 57 |
| 3.3 | Learning curves of the algorithms. | 60 |
| 3.4 | Beampattern for the HCLMS and GSC-LMS algorithms. | 61 |
| 3.5 | Beampattern for the HNCLMS and the GSC-NLMS algorithms. | 62 |
| 3.6 | Beampattern for the HCQN and the GSC-QN algorithms. | 62 |
| 4.1 | The SM-NLMS algorithm. | 68 |
| 4.2 | The SM-BNDRLMS-I algorithm: (a) The orthogonal projection onto the nearest boundary of $\mathcal{H}(k)$ lies within $\mathcal{H}(k - 1)$, i.e., $\mathbf{w}'(k) \in \mathcal{H}(k - 1)$, no further update. (b) The orthogonal projection onto the nearest boundary of $\mathcal{H}(k)$, $\mathbf{w}'(k)$, lies outside $\mathcal{H}(k - 1)$, final solution at the nearest intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k - 1)$ | 72 |
| 4.3 | Distance evolution. | 74 |
| 4.4 | General algorithm update. | 76 |
| 4.5 | The SM-BNDRLMS-II algorithm. | 78 |
| 4.6 | Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS, the BNDRLMS, and the NLMS algorithms. Condition number of the input-signal correlation matrix = 100, $SNR = 80$ dB, and $\gamma = \sqrt{5}\sigma_n$ | 87 |
| 4.7 | Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS, the BNDRLMS, and the NLMS algorithms. Condition number of the input-signal correlation matrix = 100, $SNR = 20$ dB, and $\gamma = \sqrt{5}\sigma_n$ | 88 |
| 4.8 | The overall complexity normalized with N versus the number of data points in the simulation for $SNR = 80$ dB. | 89 |
| 4.9 | The overall complexity normalized with N versus the number of data points in the simulation for $SNR = 20$ dB. | 89 |
| 4.10 | Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS. Condition number of the input-signal correlation matrix = 100, $SNR = 10$ dB, and $\gamma = \sigma_n$ | 90 |
| 4.11 | MSE for $N = 10$ as function of γ^2/σ_n^2 , for the input signals as modeled. | 92 |
| 4.12 | MSE for $N = 10$ as function of γ^2/σ_n^2 for white input signals. | 93 |

| | | |
|------|---|-----|
| 4.13 | MSE for $N = 10$ as function of γ^2/σ_n^2 for colored input signals. | 94 |
| 5.1 | General algorithm update. | 102 |
| 5.2 | Update resulting in zero <i>a posteriori</i> error. | 104 |
| 5.3 | Update resulting in constant <i>a posteriori</i> error. | 105 |
| 5.4 | Learning curves for the SM-AP algorithm with $P = 1, P = 2, P = 4, P = 6$ and the conventional AP algorithm with $P = 4, \gamma = \sqrt{5\sigma_n^2}, SNR = 80$ dB, and colored input signal. | 107 |
| 5.5 | Learning curves for the adaptive equalizer using the SM-AP algorithm with $P = 1, P = 2, P = 3$, the SM-BNDRLMS-I, and the conventional NLMS algorithm with $\mu = 1, \gamma = \sqrt{9\sigma_n^2}, SNR = 30$ dB. | 108 |
| 5.6 | Learning curves for the adaptive equalizer using the SM-AP algorithm with $P = 1, P = 2, P = 3$, the SM-BNDRLMS-I, and the conventional NLMS algorithm with $\mu = 1, \gamma = \sqrt{9\sigma_n^2}, SNR = 15$ dB. | 109 |
| 6.1 | The Householder transformation on the coefficient vector. | 115 |
| 6.2 | Geometrical interpretation of the SM-CAP algorithm in \mathbb{R}^2 for $N = 2$ and $P = 1$ | 120 |
| 6.3 | Learning curves for the CAP and the SM-CAP algorithms with $P = 1, P = 2$, and $P = 4$ data reuses, $\sigma_n^2 = 10^{-10}, \gamma = 3\sigma_n$, and colored input signal. | 124 |
| 6.4 | First experiment: no bias in the coefficient vector. | 126 |
| 6.5 | The mean output power | 127 |
| 6.6 | Coefficient-vector deviation for the second experiment. | 128 |
| 6.7 | SINR versus iteration for the CAP ($P = 1$), SM-CAP ($P = 1$), APOC-LMS, NCLMS (without correction term) algorithms, 16-bits fixed-point arithmetic, and $\gamma = (1 + 0.1)$ | 130 |
| 6.8 | Constraint deviation versus iteration for the CAP ($P = 1$), SM-CAP ($P = 1$), APOC-LMS, NCLMS (without correction term) algorithms, for 16-bits fixed-point arithmetic, and $\gamma = (1 + 0.1)$ | 131 |

| | | |
|------|---|-----|
| 7.1 | The solution $\mathbf{w}(k + 1)$ is the PU-NLMS algorithm update obtained with a time-varying step size $\mu(k) = \ \mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\ ^2/\ \mathbf{x}(k)\ ^2$, or equivalently, the M-Max NLMS algorithm with unity step size. | 139 |
| 7.2 | Geometric illustration of an update in \mathbb{R}^3 using $L = 1$ coefficient in the partial update, and with $ x(k - 2) > x(k - 1) > x(k) $, the direction of the update is along the vector $[0 \ 0 \ x(k - 2)]^T$ forming an angle θ with the input vector $\mathbf{x}(k)$ | 141 |
| 7.3 | General projection solution for $\ \mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\ ^2 \geq \alpha(k)\ \mathbf{x}(k)\ ^2$ | 142 |
| 7.4 | Projection solution with temporary expansion of the constraint set $\mathcal{H}(k)$ using a new threshold $\gamma(k)$ | 143 |
| 7.5 | Coefficient-error norm evolution. | 145 |
| 7.6 | Learning curves for the PU-NLMS algorithm for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, $SNR = 60$ dB. | 149 |
| 7.7 | Excess MSE for the PU-NLMS algorithm versus the step size μ for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, $SNR = 60$ dB. | 150 |
| 7.8 | Theoretical learning curves for different choice of step size in the PU-NLMS algorithm for $N = 51$ and $L = 10$, $SNR = 60$ dB. | 151 |
| 7.9 | Comparison of Equation (7.32) (solid lines) with the excess MSE formula obtained from literature (dashed lines). | 151 |
| 7.10 | Learning curves for the SM-PU-NLMS algorithm using fixed L for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, and the PU-NLMS algorithm for $L = 5$, $SNR = 60$ dB. | 152 |
| 7.11 | Learning curves, $L \leq L_{max}$ (dashed) and L fixed (solid). | 153 |
| 7.12 | Number of coefficients updated in the partial update versus time in a single realization for the SM-PU-NLMS algorithm with L variable and $L \leq 25$ | 154 |

Chapter 1

Introduction

1.1 Motivation

When designing optimal receiver filters for various applications, it is desirable that all unwanted interference is minimized. To implement a high-performance receiver, the receiver filter requires functions that can estimate the interference statistics. The cost of this design process could turn out to be impractical for several reasons: (1) The interference in the system can be time-varying, which is often the case in communications applications. This would require continuous retransmission of some of the interference parameters from the transmitter to the receiver. (2) The computational cost of the solution may render it impractical.

Instead of having a solution where all the interference statistics are required to be known *a priori* for the filter design, we could think of an implementation which recursively (or adaptively) estimates the interference, and as time proceeds, incorporates this knowledge into a dynamic filter design. In other words, the receiver filter would be an *adaptive filter*. One important factor contributing to the power consumption of the implementation is the computational complexity of the employed adaptive schemes. In practice, we are often faced with a penalty in terms of the receiver performance when reducing the computational complexity of the adaptive scheme (see discussions in Section 1.2 of this chapter). Therefore, an interesting topic of research is the development of low-complexity adaptive

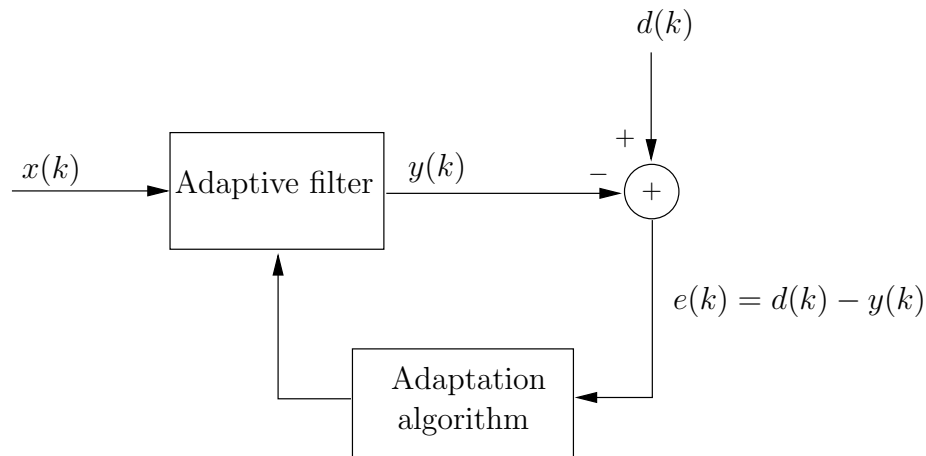


Figure 1.1: Schematic diagram of an adaptive filter.

implementations where this penalty is kept to a minimum

The thesis is concerned with the derivation of new efficient adaptive filtering algorithms. The algorithm derivations are general in the sense that they can be applied in any application where an adaptive filter may be needed. Recently adaptive solutions for communications applications have received a lot of attention in the literature. As the data-rates of evolving communications systems increases, the digital-signal processors will have less time to perform the required calculations. Therefore, communications applications, as well as other applications, will benefit from reduced-complexity adaptive filtering algorithms.

The rest of the chapter is organized as follows. First we will give a brief review of the area of adaptive filtering and give examples of its most common applications. In this way, the reader can appreciate the concepts and terms used when listing the contributions of this thesis. Furthermore, we review four adaptation algorithms which we will extensively refer to throughout the thesis. Finally at the end of the chapter, we outline in detail the contributions of each chapter in the thesis.

1.2 Adaptive Filters

An adaptive filter is useful whenever the statistics of the input signals to the filter are unknown or time varying, and the design requirements for fixed filters cannot easily be specified [1]. Examples of such applications are: system identification [2], channel equaliza-

tion/identification and interference suppression in communications systems [3, 4, 5, 6, 7, 8], and acoustic echo cancellation [9, 10]. The adaptive filter measures the output signal of the filter, and compares it to a desired output signal dictated by the true system. By observing the error between the output of the adaptive filter and the desired output signal, an adaptation algorithm updates the filter coefficients with the aim to minimize an objective function. Figure 1.1 shows the basic schematic diagram of an adaptive filter, where $x(k)$, $y(k)$, $d(k)$, and $e(k)$ are the input, output, desired output and error signals of the adaptive filter for time instant k . As can be seen from Figure 1.1, the adaptive filter is a nonlinear filter through its dependence on the input signal, although, at a given time instant it will act as a linear filter. As a consequence, the analysis of an adaptive filter can be quite involved as compared to the analysis of fixed filters. On the other hand, the self-designing feature of the adaptive filter often makes it easy to apply without requiring much *a priori* specification of the problem at hand [1].

The adaptive filter can have either finite-duration impulse response (FIR) or infinite-duration impulse response (IIR). For an adaptive FIR filter the output is obtained as a linear combination of the present and the $N - 1$ past input signal samples, N being the number of filter coefficients. An adaptive FIR filter is many times preferred over an adaptive IIR filter due to its simplicity and robustness. Furthermore, many practical problems can be accurately modeled by an FIR filter, e.g., channel identification in communications systems [5, 9]. The adaptive IIR filter can serve as a viable alternative to the FIR in applications where the required order of the adaptive filter is very high, since an IIR filter in general requires fewer filter coefficients than its FIR filter counterpart [1]. Drawbacks of the adaptive IIR filter include possible stability problems and, in certain problems, lack of a unique solution [1, 11].

As already mentioned, the adaptation algorithm tries to minimize an objective function $J_{\mathbf{w}}$, which is often related to the output error. Among the most common objective functions that are used for derivation of adaptation algorithms are:

- The mean-squared error (MSE) having $J_{\mathbf{w}} = E[e^2(k)]$;
- The least-squares (LS) having $J_{\mathbf{w}} = 1/k \cdot \sum_{i=1}^k e^2(i)$;

- The weighted least-squares (WLS) having $J_{\mathbf{w}} = \sum_{i=1}^k \lambda^{k-i} e^2(i)$.

Choosing among the many different objective functions often involves a trade-off between certain conflicting performance measures. Some of the most important performance measures related to adaptive filters are [1, 2]:

- The convergence rate, i.e., the number of algorithm iterations required to converge to the vicinity of a steady-state solution.
- The accuracy of the obtained solution as compared to the optimal obtainable solution. An often used measure is the excess MSE, or the misadjustment, which quantifies how close the adaptive filter coefficients are to the ones of the optimal filter.
- The computational complexity of the algorithm.
- Robustness to quantization when implemented in finite-precision.
- Tracking ability, i.e., the performance of the filter when operating in a nonstationary environment.

As previously stated, these performance measures are often conflicting and as a consequence, specifications on the adaptive filter in terms of these measures cannot in general be met simultaneously. For example, fast convergence rate usually implies computationally demanding implementation. On the other hand, if low misadjustment is desired, an algorithm of low computational complexity would most likely suffer from slow convergence.

1.3 Applications

This section reviews three applications where adaptive filters are frequently used. For more detailed discussion, see [1, 12].

1.3.1 System Identification

In many applications it is necessary to identify an unknown system. Examples of such applications are: identification of the acoustic echo path in acoustic echo cancellation [9,

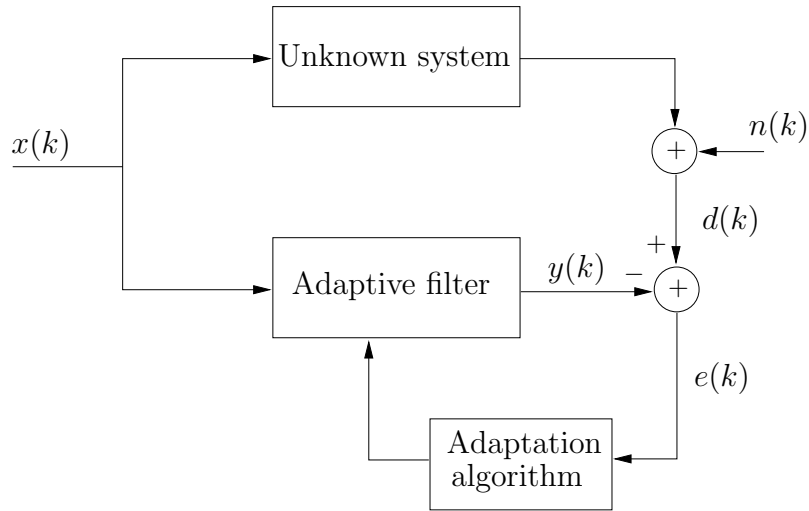


Figure 1.2: Schematic diagram of a system-identification application.

13, 14], channel identification in communications systems [8], and active noise control [15]. Figure 1.2 shows the basic structure of a system-identification application. The adaptive filter and the unknown system share the same input signal, usually a wideband signal in the case of channel identification or a noisy voice signal in the case of acoustic echo cancellation. The adaptation algorithm compares the desired signal with the output of the adaptive filter in order to minimize the chosen objective function $J_{\mathbf{w}}$. The desired signal will, in addition to the output from the unknown system, contain some measurement noise $n(k)$ which will affect the variance of the estimate of the unknown system [1].

1.3.2 Interference Suppression in Communications Systems

In wireless communications systems the main factors limiting the system capacity are various kinds of interference such as intersymbol interference (ISI) due to multipath propagation in frequency selective fading channels, cochannel (or multiple access) interference, and adjacent channel interference. ISI is the main impairment in single user communications and can be corrected through the use of an adaptive equalizer [3, 16, 17]. In multiuser communications systems, e.g., code division multiple access (CDMA) systems, the dominant source of impairment is cochannel interference coming from simultaneously active user signals occupying the same frequency band. In multi-rate CDMA systems, e.g., the

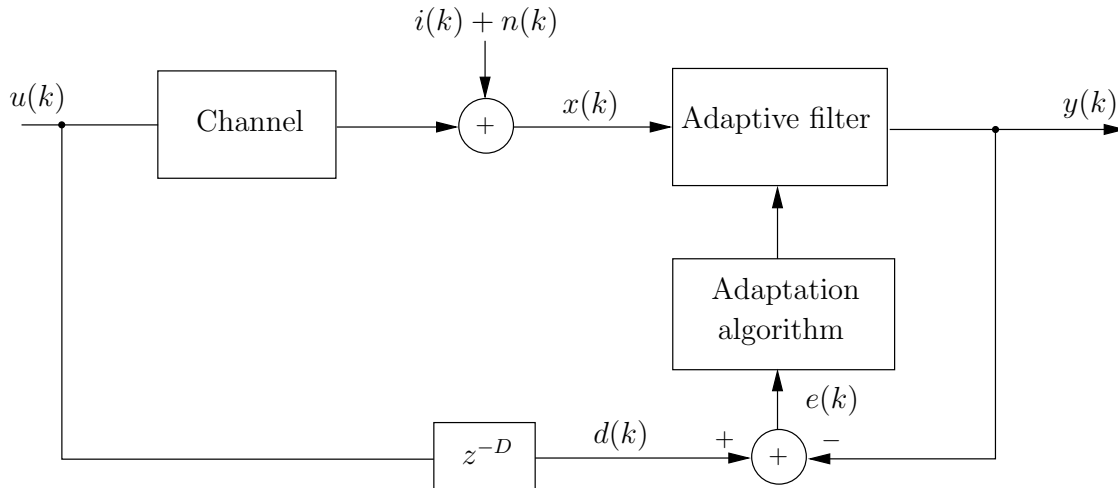


Figure 1.3: Schematic diagram of an adaptive equalizer.

third generation (3G) mobile communications systems [18], ISI suppression can also substantially improve the capacity for the high data-rate applications [19, 20, 21, 22]. There is an extensive literature dealing with multiple access (or multiuser) interference suppression in batch and in adaptive modes, see, e.g., [4, 7, 20, 23, 24, 25, 26].

The adaptive mode is very attractive due to its implementation in situations where there are tight complexity and processing-delay constraints, as in the case of a mobile handset. Adaptive cochannel and ISI suppression schemes many times resemble the single-user equalizer. Figure 1.3 shows the structure of an adaptive equalizer, where $u(k)$ is the user signal of interest and $i(k)$ is cochannel interference. The adaptive filter will try to suppress the channel-induced ISI, and in certain applications also the cochannel interference when it can be considered cyclo-stationary, e.g., in CDMA systems with short user spreading codes [25]. The desired signal $d(k)$ is now a delayed replica of the transmitted signal, where the value of the delay D is chosen to compensate for the delay introduced by the channel.

1.3.3 Antenna Arrays

In various applications, knowledge of certain signal parameters can be included in the adaptation algorithm such that the adaptive filter is constrained to provide a solution that satisfies some *a priori* knowledge. A class of such adaptive filters is the linearly constrained

minimum variance (LCMV) adaptive filter, which requires a set of linear constraints to be satisfied at each iteration. The LCMV filter finds applications in antenna array processing [6, 27, 28, 29] and blind multiuser detection in CDMA systems [25, 30, 31, 32, 33, 34, 35]. The adaptation algorithm controlling the coefficients of the adaptive filter is minimizing an objective function $J_{\mathbf{w}}$ with the additional requirement of meeting a set of linear constraints. By imposing linear constraints on the adaptive filter, the necessity of a desired signal can often be relaxed, resulting in what is commonly referred to as *blind algorithms*. The linear constraints usually reflect the prior knowledge of the system, like the direction of arrival (DOA) of user signals in antenna array processing, the user spreading code in blind multiuser detection, the linear phase requirement in system identification [36]. Figure 1.4 shows an LCMV antenna array with M antennas in a system containing two user signals, $\mathbf{u}_1(k)$ and $\mathbf{u}_2(k)$, transmitting the data $u_1(k)$ and $u_2(k)$, respectively. The user signals are impinging the array from the directions specified by the vectors $\mathbf{s}(\theta_1)$ and $\mathbf{s}(\theta_2)$, θ_i being the DOA of user i . The vector $\mathbf{s}(\theta_i)$ is referred to as the *steering vector* and in case of a uniform linear array (ULA) with element spacing of half the wavelength, the steering vector has the particular simple form given by $\mathbf{s}(\theta_i) = [1 \ e^{j\pi \sin \theta_i} \ \dots \ e^{j(M-1)\pi \sin \theta_i}]^T$ [6, 37]. The adaptive filter in this particular example is updated through the minimization of the objective function $J_{\mathbf{w}}$ under the constraint of unity response in the direction of the user of interest.

1.4 Adaptation Algorithms

This section briefly introduces four well-known algorithms that possess different qualities in terms of the performance measures mentioned in Section 1.1. The algorithms are presented here for future reference since adaptation algorithms derived within this thesis will be based on or bear similarities to these algorithms. The first algorithm to be discussed in Section 1.4.1 is the celebrated least mean-square (LMS) algorithm proposed by Widrow and Hoff [38] in 1960. Due to its low computational complexity, it still remains one of the most popular adaptive filtering algorithms. We also discuss the NLMS algorithm [39]

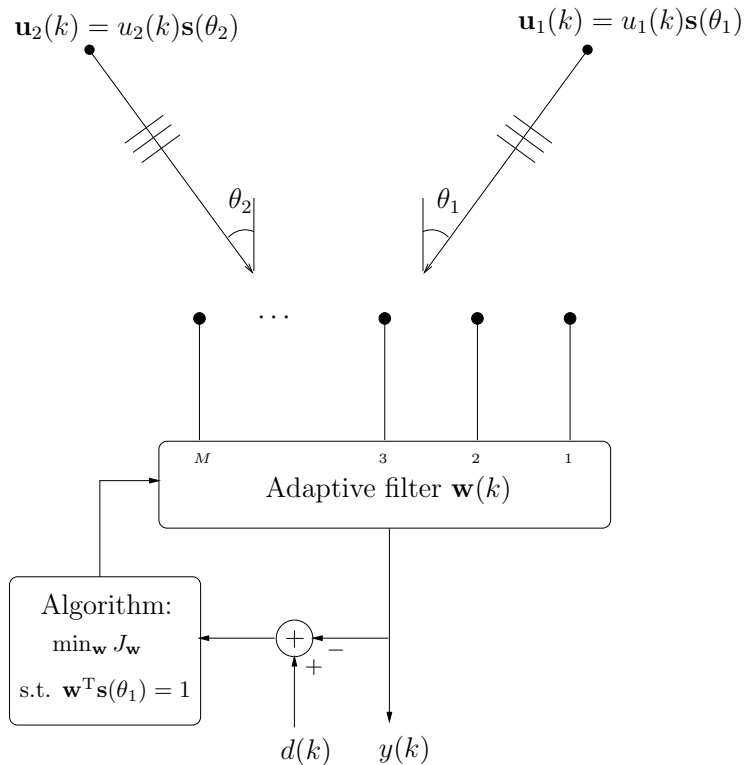


Figure 1.4: Schematic diagram of an LCMV adaptive antenna array consisting of M antennas, with an adaptation algorithm minimizing the objective function $J_{\mathbf{w}}$ while constrained to pass the desired signal from user direction $\mathbf{s}(\theta_1)$ with unity response.

proposed in 1967, which can be seen as a version of the LMS algorithm where the step size is time varying.

Thereafter, in Section 1.4.2, we review the recursive least-squares (RLS) algorithm [40] proposed in 1950, which is among the fastest adaptive filtering algorithms in terms of convergence speed. The connection between the RLS algorithm and the Kalman filter theory can be found in [12, 41]. The high computational complexity of the RLS algorithm can be significant in applications where the order of the adaptive filter is high. This inspired the development of algorithms with computational complexity somewhere in between those of the LMS and RLS algorithms. The affine-projection (AP) algorithm [42] presented in Section 1.4.3 utilizes the concept of reusing past information to improve the convergence speed. Finally, the quasi-Newton (QN) algorithm [43] is discussed in Section 1.4.4. The QN algorithm performs favorably in finite-precision as compared to the RLS algorithm.

The adaptive filter considered in this thesis is assumed to be of FIR type with N

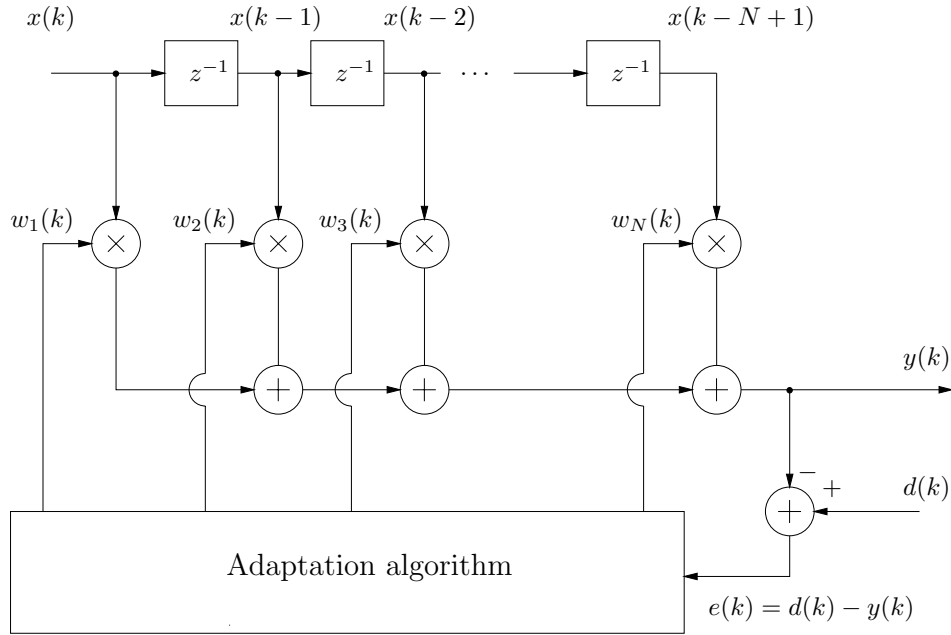


Figure 1.5: Adaptive FIR filter.

coefficients such that the output of the adaptive filter can be expressed as

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \quad (1.1)$$

where

$$\mathbf{w}(k) = [w_1(k) \ w_2(k) \ \cdots \ w_N(k)]^T \quad (1.2)$$

is the vector containing the coefficients of the adaptive filter, and

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \cdots \ x(k-N+1)]^T \quad (1.3)$$

is the vector containing the input samples. The structure of the adaptive filter is shown in Figure 1.5. In the particular application of antenna arrays discussed in Section 1.3.3, the input vector is different but the algorithms to be discussed below are the same.

1.4.1 The Least Mean-Square (LMS) Algorithm

The least mean-square (LMS) algorithm [1, 12, 38] is probably the most widely used adaptive filtering algorithm, being employed in several communications systems. It has gained

Table 1.1: The Least Mean-Square Algorithm

| LMS Algorithm |
|---|
| for each k { $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k)$ } |

popularity due to its low computational complexity and proven robustness. The LMS algorithm is a gradient-type algorithm that updates the coefficient vector by taking a step in the direction of the negative gradient [12] of the objective function, i.e.,

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\mu}{2} \frac{\partial J_{\mathbf{w}}}{\partial \mathbf{w}(k)} \quad (1.4)$$

where μ is the step size controlling the stability, convergence speed, and misadjustment. To find an estimate of the gradient, the LMS algorithm uses as objective function the instantaneous estimate of the MSE, i.e., $J_{\mathbf{w}} = e^2(k)$ resulting in the gradient estimate $\partial J_{\mathbf{w}}/\partial \mathbf{w}(k) = -2e(k)\mathbf{x}(k)$ [12]. The pseudo-code for the LMS algorithm is shown in Table 1.1. In order to guarantee stability in the mean-squared sense, the step size μ should be chosen in the range $0 < \mu < 2/\text{tr}\{\mathbf{R}\}$, where $\text{tr}\{\cdot\}$ is the trace operator and

$$\mathbf{R} = \text{E}[\mathbf{x}(k)\mathbf{x}^T(k)]$$

is the input-signal autocorrelation matrix. The upper bound should be considered optimistic and in practice a smaller value is recommended [1]. A normalized version of the LMS algorithm, the NLMS algorithm [1, 39, 44], is obtained by substituting the step size in Equation (1.4) with the time-varying step size $\mu/\|\mathbf{x}(k)\|^2$, where $0 < \mu < 2$ [44]. The NLMS algorithm is in the control literature referred to as the *projection algorithm* (PA) [45]. The main drawback of the LMS and the NLMS algorithms is the slow convergence for colored noise input signals. In cases where the convergence speed of the LMS algorithm is not satisfying, the adaptation algorithms presented in the following sections may serve as viable alternatives.

Table 1.2: The Recursive Least-Squares Algorithm

| RLS Algorithm |
|---|
| $\mathbf{R}^{-1}(0) = \delta^{-1}\mathbf{I}$, δ small positive constant for each k { $\mathbf{k}(k) = \mathbf{R}^{-1}(k-1)\mathbf{x}(k)$ $\boldsymbol{\kappa}(k) = \frac{\mathbf{k}(k)}{\lambda + \mathbf{x}^T(k)\mathbf{k}(k)}$ $\mathbf{R}^{-1}(k) = \frac{1}{\lambda} \left[\mathbf{R}^{-1}(k-1) - \frac{\mathbf{k}(k)\mathbf{k}^T(k)}{\lambda + \mathbf{x}^T(k)\mathbf{k}(k)} \right]$ $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ $\mathbf{w}(k+1) = \mathbf{w}(k) + e(k)\boldsymbol{\kappa}(k)$ } |

1.4.2 The Recursive Least-Squares (RLS) Algorithm

To overcome the problem of slow convergence of the LMS algorithm operating in colored environment, one can implement the recursive least-squares (RLS) algorithm [1, 12]. The RLS algorithm is a recursive implementation of the least-squares (LS) solution, i.e., it minimizes the LS objective function. The recursions for the most common version of the RLS algorithm, which is presented in its standard form in Table 1.2, is a result of the weighted least-squares (WLS) objective function $J_{\mathbf{w}} = \sum_{i=1}^k \lambda^{k-i} e^2(i)$. Differentiating the objective function $J_{\mathbf{w}}$ with respect to $\mathbf{w}(k)$ and solving for the minimum yields the following equations

$$\left[\sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i)\mathbf{x}^T(i) \right] \mathbf{w}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i)d(i) \quad (1.5)$$

where $0 < \lambda \leq 1$ is an exponential scaling factor often referred to as the forgetting factor.

Defining the quantities

$$\mathbf{R}(k) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i)\mathbf{x}^T(i)$$

and

$$\mathbf{p}(i) = \sum_{i=1}^k \lambda^{k-i} \mathbf{x}(i)d(i),$$

the solution is obtained as

$$\mathbf{w}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k).$$

The recursive implementations is a result of the formulations

$$\mathbf{R}(k) = \lambda \mathbf{R}(k-1) + \mathbf{x}(k)\mathbf{x}^T(k)$$

and

$$\mathbf{p}(k) = \lambda \mathbf{p}(k-1) + \mathbf{x}(k)d(k).$$

The inverse $\mathbf{R}^{-1}(k)$ can be obtained recursively in terms of $\mathbf{R}^{-1}(k-1)$ using the *matrix inversion lemma*¹ [1] thus avoiding direct inversion of $\mathbf{R}(k)$ at each time instant k . The main problems with the RLS algorithm are potential divergence behavior in finite-precision environment and high computational complexity, which is of order N^2 . The stability problems are usually a result of lost symmetry and positive definiteness of the matrix $\mathbf{R}^{-1}(k)$. More robust implementations exist based on square-root factorization or QR decomposition of matrix $\mathbf{R}^{-1}(k)$, see, for example, [1, 12, 46]. Various versions of the so-called fast transversal algorithms with computational complexity of order N have been proposed [47, 48, 49, 50] but many of these suffer from stability problems when implemented in finite precision.

Algorithms whose convergence rate and computational complexity are somewhere between those of the LMS and RLS algorithms are considered in the following section.

1.4.3 The Affine-Projection (AP) Algorithm

It is well known that *normalized* LMS algorithms often converge faster than the basic LMS algorithm, and can many times provide a viable alternative to the RLS algorithm. Examples of such low-complexity algorithms are the binormalized data-reusing least mean-square (BNDRLMS) [51, 52], the normalized new data-reusing (NNDR) [53], and the affine-projection (AP) [13, 42, 54, 55, 56, 57] algorithms. The idea of re-utilizing past and present information in the coefficient-update, called data-reusing, has shown to be a promising approach to balance convergence speed and computational complexity of the algorithm. The BNDRLMS algorithm utilizes current and past data-pairs in its update. The relationships

¹ $[\mathbf{A} + \mathbf{BCD}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}[\mathbf{DA}^{-1}\mathbf{B} + \mathbf{C}^{-1}]^{-1}\mathbf{DA}^{-1}$

Table 1.3: The Affine-Projection Algorithm

| AP Algorithm |
|---|
| for each k { $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k)$ $\mathbf{t}(k) = [\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}]^{-1} \mathbf{e}(k)$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{X}(k)\mathbf{t}(k)$ } |

between a number of data-reusing algorithms were addressed in [58]. The AP algorithm can be seen as a general normalized data-reusing algorithm that reuses an arbitrary number of data-pairs.

The AP algorithm updates its coefficient vector such that the new solution belongs to the intersection of P hyperplanes defined by the present and the $P - 1$ previous data pairs $\{\mathbf{x}(i), d(i)\}_{i=k-P+1}^k$. The optimization criterion used for the derivation of the AP algorithm is given by

$$\begin{aligned} \mathbf{w}(k+1) &= \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to} \\ \mathbf{d}(k) &= \mathbf{X}^T(k)\mathbf{w} \end{aligned} \quad (1.6)$$

where

$$\begin{aligned} \mathbf{d}(k) &= [d(k) \ d(k-1) \ \cdots \ d(k-P+1)]^T \\ \mathbf{X}(k) &= [\mathbf{x}(k) \ \mathbf{x}(k-1) \ \cdots \ \mathbf{x}(k-P+1)] \end{aligned} \quad (1.7)$$

The updating equations for the AP algorithm obtained as the solution to the minimization problem in (1.6) are presented in Table 1.3 [13, 42]. To control stability, convergence, and final error, a step size μ is introduced where $0 < \mu < 2$ [54]. To improve robustness a diagonal matrix $\delta\mathbf{I}$ (δ a small constant) is used to regularize the inverse matrix in the AP algorithm.

1.4.4 The Quasi-Newton (QN) Algorithm

The RLS algorithm with its fast convergence relies on the estimation of the inverse of the correlation matrix $\mathbf{R}^{-1}(k)$. For stability, it is required that $\mathbf{R}^{-1}(k)$ remains symmetric and positive definite. However, implementation in finite precision may cause $\mathbf{R}^{-1}(k)$ to become indefinite [43]. One algorithm that provides convergence speed comparable to that of the RLS algorithm but is guaranteed to be stable even under high input-signal correlation and fixed-point short-wordlength arithmetic is the quasi-Newton (QN) algorithm [43, 59].

In the QN algorithm, the coefficient vector is updated as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\mathbf{h}(k) \quad (1.8)$$

where $\mu(k)$ is a step size obtained through an exact line search, and $\mathbf{h}(k)$ is the direction of update given by

$$\mathbf{h}(k) = -\mathbf{R}^{-1}(k-1) \frac{\partial J_{\mathbf{w}}}{\partial \mathbf{w}(k)} \quad (1.9)$$

Choosing $J_{\mathbf{w}} = e^2(k)$ we have $\partial J_{\mathbf{w}} / \partial \mathbf{w}(k) = -2e(k)\mathbf{x}(k)$, and performing an exact line search results in a step size [43]

$$\mu(k) = \frac{1}{2\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)} \quad (1.10)$$

The update of $\mathbf{R}^{-1}(k)$ is crucial for the numerical behavior of the QN algorithm, and different approximations lead to different QN algorithms. In [43] an approximation of $\mathbf{R}^{-1}(k)$ was given that is robust and remains positive definite even for highly correlated input signals and short wordlength arithmetic

$$\mathbf{R}^{-1}(k) = \mathbf{R}^{-1}(k-1) + [\mu(k) - 1] \frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)}{\mathbf{x}^T(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)} \quad (1.11)$$

where the initial value $\mathbf{R}^{-1}(0)$ can be any positive definite matrix, usually chosen as $\mathbf{R}^{-1}(0) = \delta\mathbf{I}$ with $\delta > 0$.

The QN algorithm can be implemented as shown in Table 1.4. In Table 1.4 a positive constant α is used to control the speed of convergence and the misadjustment. Convergence in the mean and the mean-squared sense of the coefficient vector is guaranteed for $0 < \alpha < 2$ provided that $\mathbf{R}^{-1}(k)$ is positive definite [43, 59, 60].

Table 1.4: The Quasi-Newton Algorithm

| QN Algorithm |
|--|
| for each k { $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ $\mathbf{t}(k) = \mathbf{R}^{-1}(k-1)\mathbf{x}(k)$ $\tau(k) = \mathbf{x}^T(k)\mathbf{t}(k)$ $\mu(k) = \frac{1}{2\tau(k)}$ $\mathbf{R}^{-1}(k) = \mathbf{R}^{-1}(k-1) + \frac{[\mu(k)-1]}{\tau(k)}\mathbf{t}(k)\mathbf{t}^T(k)$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha \frac{e(k)}{\tau(k)}\mathbf{t}(k)$ } |

1.4.5 Complex-Valued Signals

The algorithms in this thesis are derived for real-valued input signals. The extension of the algorithms to work with complex-valued input signals is straightforward [1, 12, 61, 62].

In the case of complex-valued input signals, let $J_{\mathbf{w}, \mathbf{w}^*}$ denote the real-valued objective function of the weight vector \mathbf{w} to be solved for, and \mathbf{w}^* denotes the conjugate of vector \mathbf{w} . The maximum rate of change of $J_{\mathbf{w}, \mathbf{w}^*}$ is given by $\partial J_{\mathbf{w}, \mathbf{w}^*} / \partial \mathbf{w}^*$ [61, 62]. In order to get a meaningful result, the objective function needs to have explicit dependency on the conjugate of the weight vector. Usually this simply translates into changing transposition to conjugate transposition (or Hermitian). For a more detailed discussion on the topic, see [61, 62].

To illustrate the procedure of deriving the complex version of an algorithm, let us take as an the complex-valued LMS algorithm discussed in Section 1.4.1. The objective function becomes $J_{\mathbf{w}, \mathbf{w}^*} = |e(k)|^2$, where $e(k) = d(k) - \mathbf{w}^H \mathbf{x}(k)$. Therefore we have

$$\begin{aligned} \frac{\partial J_{\mathbf{w}, \mathbf{w}^*}}{\partial \mathbf{w}^*} &= \frac{\partial \{ [d(k) - \mathbf{w}^H \mathbf{x}(k)]^H [d(k) - \mathbf{w}^H \mathbf{x}(k)] \}}{\partial \mathbf{w}^*} \\ &= -e^*(k) \mathbf{x}(k) \end{aligned} \tag{1.12}$$

As with the LMS algorithm for real-valued input signals, the update is obtained by taking a small step in the negative gradient of the objective function. With the above definition

of the gradient, we get

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e^*(k)\mathbf{x}(k) \quad (1.13)$$

1.5 Overview and Contributions

The scope of this thesis is to develop new adaptation algorithms for both unconstrained adaptive filters and LCMV adaptive filters as depicted in Figures 1.1 and 1.4, respectively. The focus of the thesis is on FIR adaptive filters. The performance of new algorithms is evaluated through analysis, and simulations in the applications described in Section 1.3. Three strategies, to be described shortly, are used to derive adaptation algorithms with low computational complexity: (1) rank reduction of linearly constrained filters through a transformation of the input signal; (2) application of the set-membership filtering (SMF) framework, and; (3) application of partial-update.

Below we give a brief background of each strategy:

- (1) In LCMV adaptive filtering, the computational complexity is not only due to the adaptation algorithm employed. The adaptive LCMV filter updates in a subspace orthogonal to the space spanned by the set of linear constraints. Therefore, the adaptation algorithms usually make use of a projection matrix, and the form of this matrix affects the overall computational complexity of the implementation. A method to reduce the computational complexity for the implementation of LCMV adaptive filters is to employ reduced-rank updating [30, 63, 64, 65, 66, 67] where a transformation is applied to the input signal to reduce the dimension of the problem such that adaptation can be performed in a reduced subspace.
- (2) Adaptive SMF [68, 69, 70, 71, 72] is a recent approach to adaptive filtering. The SMF framework specifies a bound on the output estimation error, and as a consequence, there exists a set of feasible solutions. The adaptive SMF only performs updates if the magnitude of the error exceeds the predefined bound and, therefore, a reduction of the average computational complexity is obtained.

- (3) Partial-update (PU) adaptive filtering is a technique suitable for applications where the order of the adaptive filter is so high that it may even prohibit the use of the NLMS algorithm, see, e.g., [14, 73, 74]. PU adaptive filters reduce the algorithm complexity by properly decreasing the number of filter coefficients that are updated at each iteration so that the filter order may be kept fixed.

The chapters in the thesis are arranged as follows. Chapter 2 provides new results on the equivalence of constrained adaptive filtering structures together with the introduction of a new adaptation algorithm. Furthermore, an introduction to the field of linearly constrained adaptive filters is provided to give the necessary background for the new approach introduced in Chapter 3. Chapter 3 derives new constrained adaptive filtering algorithms where rank reduction is performed through an orthogonal transformation of the input signal. Chapter 4 introduces and analyzes novel data-selective normalized adaptive filtering algorithms with two data-reuses, and Chapter 5 extends this work to include an arbitrary number of data reuses. Chapter 6 derives constrained affine projection algorithms and reduced computational complexity schemes are obtained through both rank reduction and application of the set-membership filtering framework. Finally, Chapter 7 investigates and analyzes partial-update adaptive filters. Furthermore, partial-update is combined with the framework of set-membership filtering to derive novel low-complexity adaptation algorithms.

Table 1.5 shows how the algorithms developed in each chapter are related to different approaches to adaptive filtering. Furthermore, it provides information on how the different fields can be combined to obtain efficient adaptation algorithms.

The scientific contributions of the thesis are found in Chapters 2–7. Chapter 2 also reviews known results from the literature in order to provide necessary background material for Chapter 3. The main contributions of Chapter 2 can be found in Sections 2.3 and 2.5. In Chapters 3–7 we have tried to keep the necessary background material to a minimum and it is mostly appearing in the introductory part together with relevant references. In particular, Chapter 4.2–4.2.1 contains background material on set-membership adaptive filtering, which is extensively used for the derivation of the algorithms through

Table 1.5: Characterization of the Algorithms Developed in the Thesis.

| Chapter | LCMV | Data-Reusing | Set-Membership Filtering | Partial-Update |
|---------|------|--------------|--------------------------|----------------|
| 2 | × | | | |
| 3 | × | | | |
| 4 | | × | × | |
| 5 | | × | × | |
| 6 | × | × | × | |
| 7 | | | × | × |

Chapters 4–7. In order to clarify the contributions of the thesis, they are listed below for each chapter.

Chapter 2: Constrained Adaptive Filters

- A new LCMV adaptive filtering algorithm is proposed, namely, the normalized constrained LMS (NCLMS).
- Equivalence study of the transients of the constrained RLS (CRLS) algorithm and the generalized sidelobe canceller (GSC) structure [28] employing an RLS algorithm. It is shown that the two implementations produce identical transients assuming proper initializations.

Chapter 3: Householder Constrained Adaptation Algorithms

- An efficient implementation of LCMV adaptive filters is presented based on Householder transformation of the input signal vector. The transformation of the input signal reduces the dimension of the subspace in which the adaptive-filter coefficients are updated. The approach allows application of any unconstrained adaptation algorithm to a linearly constrained problem, and always renders efficient and robust implementations.
- Derivation of several adaptation algorithms based on Householder transformations such as the constrained versions of the LMS, the NLMS, and the QN algorithms.

Chapter 4: Set-Membership Binormalized Data-Reusing LMS Algorithms

- Transient analysis of the SM-NLMS algorithm [69] is performed, indicating a decrease in convergence speed for correlated input signals.
- Two new algorithms, called set-membership binormalized LMS (SM-BNDRLMS) algorithms, are derived using the concept of set-membership filtering (SMF). They can be regarded as a generalization of the recently proposed set-membership NLMS (SM-NLMS) algorithm [69]. They reuse two past input and desired signal pairs in order to construct a space of feasible solutions for the coefficient updates. The algorithms include data-dependent step sizes that provide fast convergence and low excess MSE. Convergence analyses for the mean-squared error have been done and closed-form expressions are given for both white and colored input signals.

Chapter 5: Set-Membership Affine-Projection Algorithms

- The set-membership affine-projection (SM-AP) algorithm is introduced. The algorithm generalizes the ideas of both the SM-NLMS and the SM-BNDRLMS algorithms to reuse information from several past input and desired signal pairs. The resulting algorithm can be seen as a set-membership version of the affine-projection (AP) algorithm with an optimized step size. Unlike most adaptive filtering algorithms, the SM-AP algorithm does not trade off convergence speed with misadjustment and computational complexity.

Chapter 6: Low-Complexity Constrained Affine-Projection Algorithms

- This chapter introduces the constrained affine-projection (CAP) algorithm. Analysis of the bias is provided together with an efficient implementation obtained via a unitary transformation of the input signals.
- To achieve an algorithm with a reduced computational complexity, the set-membership constrained affine-projection (SM-CAP) algorithm is derived. The algorithm updates in a way that the filter coefficients remain in a set described by both a bounded error constraint and a set of linear constraints.

- Analysis of the excess MSE is provided for the partial-update NLMS (PU-NLMS) algorithm [75] and a closed-form expression is provided for the case of white noise input signals. New and more accurate bounds on the step size are provided to ensure stability of the algorithm.
- Derivation of a set-membership algorithm with partial updating. The new algorithm combines the sparse updates coming from the set-membership framework with the ideas of partial-update algorithms. Convergence in the mean-squared is shown for the case of white input signals.

Although this thesis is presented as a monograph, results have been published in journals and conference proceedings [76, 66, 67, 77, 78, 79, 80, 81, 82]. Part of the material has been submitted to journals but has not yet been accepted for publication, however, in many cases a related conference paper containing the ideas has been published.

Chapter 2

Constrained Adaptive Filters

This chapter presents a tutorial of linearly-constrained minimum-variance (LCMV) filtering. It also provides new results for the comparison of two well-known structures for implementation. The chapter serves as a necessary background for the new approach proposed in Chapter 3. Through a graphical description of the algorithms further insight on linearly constrained adaptive filters is made possible and the main differences among several algorithms are highlighted. The related structure for LCMV filters, the generalized sidelobe canceler (GSC), is discussed as well as conditions for which the optimal and adaptive implementations of the LCMV filters and the GSC are equivalent. The main contribution of the chapter is the study of transient equivalence of the constrained RLS (CLRS) algorithm and the GSC structure employing an RLS algorithm. We prove that the two adaptive implementations are equivalent everywhere regardless of the blocking matrix chosen. This guarantees that algorithm tuning is not affected by the blocking matrix. This result differs from the more restrictive case for transient-equivalence of the constrained LMS (CLMS) algorithm and the GSC employing LMS algorithm, because in this case the blocking matrix needs to be unitary.

2.1 Introduction

Adaptive receiving-antenna systems that can operate in real time were developed in the sixties [83, 84] and were intended to perform directional and spatial filtering with minimum

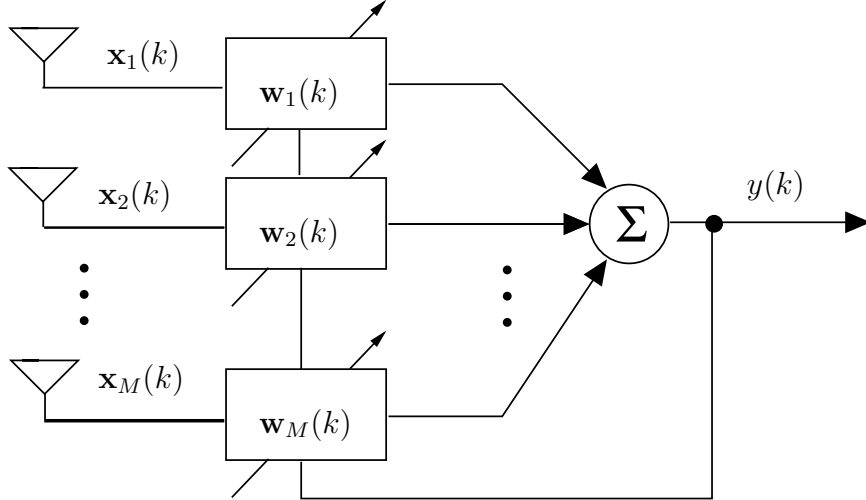


Figure 2.1: Broadband adaptive receiving array.

knowledge of the statistics of arriving signals. Linearly-constrained (LC) adaptive array processing [27] was undoubtedly a significant improvement to previously devised adaptive antenna-array systems, for the need of training sequences and knowledge of interfering-signal statistics became unnecessary. In the approach presented in [27] output power is minimized while a desired signal arriving at a known direction is linearly filtered according to a specified frequency response.

Figure 2.1 shows a schematic diagram of a broadband array-processing filter with M sensors and filters with N taps. The output of the array may be expressed as $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$ where

$$\mathbf{w}(k) = [\mathbf{w}_1^T(k) \ \mathbf{w}_2^T(k) \ \cdots \ \mathbf{w}_M^T(k)]^T \quad (2.1)$$

$$\mathbf{x}(k) = [\mathbf{x}_1^T(k) \ \mathbf{x}_2^T(k) \ \cdots \ \mathbf{x}_M^T(k)]^T \quad (2.2)$$

and

$$\mathbf{x}_i(k) = [x_i(k) \ x_i(k-1) \ \cdots \ x_i(k-N+1)]^T \quad (2.3)$$

For LC adaptive filters, coefficient update is performed in a subspace which is orthogonal to the subspace spanned by a constraint matrix [27]. Direction of update is given by the input-signal vector premultiplied by a projection matrix, which is rank-deficient. This

may be regarded as the use of nonpersistently exciting input signal,¹ and lack of persistence of excitation requires that a correction term be added to the coefficients to prevent accumulation of roundoff errors [27, 28]. The constrained LMS (CLMS) algorithm, which does not require re-initialization and incorporates the constraints into the solution was first introduced by Frost [27]. More recently, other constrained adaptation algorithms were introduced which are tailored to specific applications or present advantageous performance regarding convergence and robustness, see, e.g., [66, 67, 85, 86, 87, 88, 79]. The constrained RLS (CRLS) algorithm introduced in [85] is one solution which tries to overcome the problem of slow convergence experienced with the CLMS algorithm in situations where the input signal is strongly correlated.

An alternative approach to the implementation of LC array processing was introduced by Griffiths and Jim in [28], which became known as the generalized sidelobe canceling (GSC) model. With the GSC model, dimension of the adaptation subspace is properly reduced by means of a *blocking matrix* such that the persistence of excitation is not lost due to imposing constraints. By transforming the constrained minimization problem into an unconstrained minimization problem, the GSC model allows any adaptation algorithm be directly applied. Furthermore, as the restriction imposed on the blocking matrix is only that its columns must be orthogonal to the constraint matrix, a myriad of possible implementations result. It has been shown in [28] that, in order for the transients of the CLMS algorithm and the GSC employing an LMS algorithm to bear any relation, the blocking matrix needs to be orthogonal. An equivalence comparison of the transients had not yet been performed for the CRLS algorithm and the GSC structure employing an RLS algorithm, herein referred to as the GSC-RLS algorithm.

The rest of the chapter is organized as follows. Section 2.2 gives the background of the LCMV filter. Section 2.3 review the CLMS and the CRLS algorithms, and introduces the normalized constrained LMS (NCLMS) algorithm. Graphical descriptions of various algorithms are presented in order to provide a better understanding of the updating process used in constrained algorithms. Section 2.4 gives the background of the GSC model.

¹For a discussion on persistence of excitation, see [45].

Section 2.5 discusses the transient equivalence of the adaptive implementations of LC and GSC filters, and the CRLS and the GSC employing an RLS algorithm in particular. We try to answer the following question: Is the requirement of orthogonal blocking matrix related to the implementation of the LMS algorithm carried over to the case of the RLS algorithm? It is shown that this is not the case, and that the transients of the CRLS and the GSC-RLS algorithm are identical everywhere if the blocking matrix is orthogonal to the constraint matrix, which is the case for any GSC structure. As a consequence, any valid blocking matrix used in the GSC-RLS structure will always produce the same curves as the CRLS algorithm. Section 2.6 discusses techniques to construct the blocking matrix in the GSC model. Finally, Section 2.7 contains simulations followed by conclusions in Section 2.8.

2.2 Optimal Linearly-Constrained Minimum-Variance Filter

The optimal (LCMV) filter is the one that minimizes the objective function $J_{\mathbf{w}}$ subject to a set of linear constraints, i.e.,

$$\mathbf{w} = \arg \min_{\mathbf{w}} J_{\mathbf{w}} \quad \text{subject to} \quad \mathbf{C}^T \mathbf{w} = \mathbf{f} \quad (2.4)$$

where \mathbf{w} , as remarked before, is a vector of coefficients of length MN , \mathbf{C} is the $MN \times p$ constraint matrix, and \mathbf{f} is the $p \times 1$ gain vector, p being the number of constraints. The most common LCMV filter used in the literature is probably the one minimizing the mean output energy (MOE) objective function:

$$J_{\mathbf{w}} = \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (2.5)$$

where \mathbf{R} is the $MN \times MN$ autocorrelation matrix of the input signal. By using the method of Lagrange multipliers, the optimal solution becomes [12, 27]

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{C} (\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C})^{-1} \mathbf{f} \quad (2.6)$$

The LCMV filter in (2.6) has been used in several applications in communications systems, and its popularity may be accredited the possibility of adaptive implementations which do not require training signals, see, e.g., [25, 27, 28, 30, 31, 32]. A more general setup uses the MSE, $J_{\mathbf{w}} = E[e^2(k)] = E[\{d(k) - \mathbf{w}^T \mathbf{x}(k)\}^2]$, as the objective function, giving the optimal solution [89]

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p} + \mathbf{R}^{-1} \mathbf{C} [\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C}]^{-1} [\mathbf{f} - \mathbf{C}^T \mathbf{R}^{-1} \mathbf{p}] \quad (2.7)$$

where \mathbf{p} is the cross-correlation vector between the input vector $\mathbf{x}(k)$ and the desired signal $d(k)$. It is easy to verify that, in the absence of a desired signal, $d(k) = 0$ and $\mathbf{p} = \mathbf{0}$, the optimal solution in (2.7) becomes equal to (2.6). The rest of the chapter treats the more general case when a desired signal is present, since applications where no desired signal exists can easily be modeled by setting $d(k) = 0$. An application where a desired signal is present was discussed in [36] where constraints were used to ensure linear phase filters. Furthermore, in many wireless systems a training signal is periodically retransmitted to aid estimation of time-varying parameters, e.g., multipath channel coefficients, and in such applications we could think of a solution that switch in time between the two cases discussed above.

Both equations (2.6) and (2.7) above bear the difficulty of knowing in real-time the inverse of the input-signal autocorrelation matrix, \mathbf{R}^{-1} , and the cross-correlation vector \mathbf{p} . A much more practical approach is to produce an estimate of \mathbf{w}_{opt} recursively at every iteration. As time proceeds, the estimate is improved such that convergence in the mean to the optimal solution may eventually be achieved.

2.3 LC Adaptive Filtering

2.3.1 The Constrained Least Mean-Square Algorithm

Frost [27] has proposed an algorithm to estimate \mathbf{w}_{opt} based on the gradient method or, more specifically, based on the least-mean-square (LMS) algorithm for adaptive filtering.

Let $\mathbf{w}(k)$ denote the estimate of \mathbf{w}_{opt} at time instant k and

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \quad (2.8)$$

denote the filter output, equal in absolute value to the output error, in applications where the reference signal is zero.

The Constrained LMS (CLMS) algorithm [27] uses as an estimate of the input-signal autocorrelation matrix \mathbf{R} , at instant k , the outer product of the input-signal vector by itself, i.e., $\hat{\mathbf{R}} = \mathbf{x}(k)\mathbf{x}^T(k)$. In this case, the coefficient-update equation becomes [27]:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu e(k) [\mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T] \mathbf{x}(k) + \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)] \\ &= \mathbf{w}(k) + \mu e(k)\mathbf{P}\mathbf{x}(k) + \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)] \\ &= \mathbf{P} [\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)] + \mathbf{F} \end{aligned} \quad (2.9)$$

where \mathbf{I} is the MN th-order identity matrix [27],

$$\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \quad (2.10)$$

is the projection matrix onto the subspace orthogonal to the subspace spanned by the constraint matrix, and [27]

$$\mathbf{F} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} \quad (2.11)$$

Note that in (2.9) the term multiplied by the projection matrix, $\mathbf{w}(k) + \mu y(k)\mathbf{x}(k)$, corresponds to the unconstrained LMS solution which is projected onto the homogeneous hyperplane $\mathbf{C}^T\mathbf{w} = \mathbf{0}$ and moved back to the constraint hyperplane by adding vector \mathbf{F} . Figure 2.2 illustrates this operation.

A normalized version of the CLMS algorithm, namely the NCLMS algorithm, can be derived [86]; the update equation becomes:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P} \left[\mathbf{w}(k) + \mu \frac{e(k)}{\mathbf{x}^T(k)\mathbf{P}\mathbf{x}(k)} \mathbf{x}(k) \right] + \mathbf{F} \\ &= \mathbf{w}(k) + \mu \frac{e(k)}{\mathbf{x}^T(k)\mathbf{P}\mathbf{x}(k)} \mathbf{P}\mathbf{x}(k) + \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}(k)] \end{aligned} \quad (2.12)$$

We shall stress here the fact that for the normalized constrained LMS (NCLMS) algorithm in (2.12) the *a posteriori* output signal is zero for $\mu = 1$. The solution is at

the intersection of the hyperplane defined by the constraints, $\mathcal{H}_1 : \mathbf{C}^T \mathbf{w} = \mathbf{f}$, with the hyperplane defined by the null *a posteriori* condition $\mathcal{H}_0 : \mathbf{x}^T(k) \mathbf{w} = d(k)$. Therefore, the solution $\mathbf{w}(k+1)$ is not merely a projection of the solution of the normalized LMS (NLMS) algorithm onto the hyperplane defined by the constraints. This is also illustrated in Figure 2.2, where we present a detailed graphical description of the coefficient update of several algorithms in the case of two coefficients only. In this case, hyperplanes \mathcal{H}_0 and \mathcal{H}_1 become two lines as noted in the figure. As $\mathbf{w}(k)$ must satisfy the constraints, it must belong to \mathcal{H}_1 and can be decomposed in two mutually orthogonal vectors, \mathbf{F} and $\mathbf{P}\mathbf{w}(k)$. The figure also illustrates how the solutions of the constrained version of the LMS algorithm, the NLMS algorithm, and the projection algorithm [90] relate. Note that in this figure all updated vectors for the constrained algorithms are located along the direction of $\mathbf{C}^T \mathbf{w} = \mathbf{f}$ (points 4, 6, and 7). Therefore, if \mathbf{w} was rotated with an angle θ such that w_1 axis and \mathbf{F} had the same direction, the component along this direction would not need to be updated. This fact will be used in Chapter 3 where new Householder-Transform algorithms are introduced.

The necessity of the last term in (2.9) and (2.12) may be surprising, for it is expected that all $\mathbf{w}(k)$ satisfy the constraint and, therefore, this last term should be equal to zero. In practical implementations, however, this term shall be included to prevent divergence in a limited-precision arithmetic machine [28] due to perturbations introduced in the coefficient vector in a direction not excited by vector $\mathbf{P}\mathbf{x}(k)$. The same reasoning can be applied to the constrained recursive least-squares (CRLS) algorithm presented in [85] and to the constrained quasi-Newton (CQN) algorithm presented in [87].

2.3.2 The Constrained Least-Squares Algorithm

The constrained recursive least-squares (CRLS) algorithm to be discussed below uses the weighted least-squares criterion, $J_{\mathbf{w}} = \sum_{i=1}^k \lambda^{k-i} e^2(i)$, as an objective function resulting in the following optimization problem

$$\mathbf{w}(k) = \arg \min_{\mathbf{w}} \{ \mathbf{e}^T(k) \mathbf{e}(k) \} \quad \text{subject to} \quad \mathbf{C}^T \mathbf{w} = \mathbf{f} \quad (2.13)$$

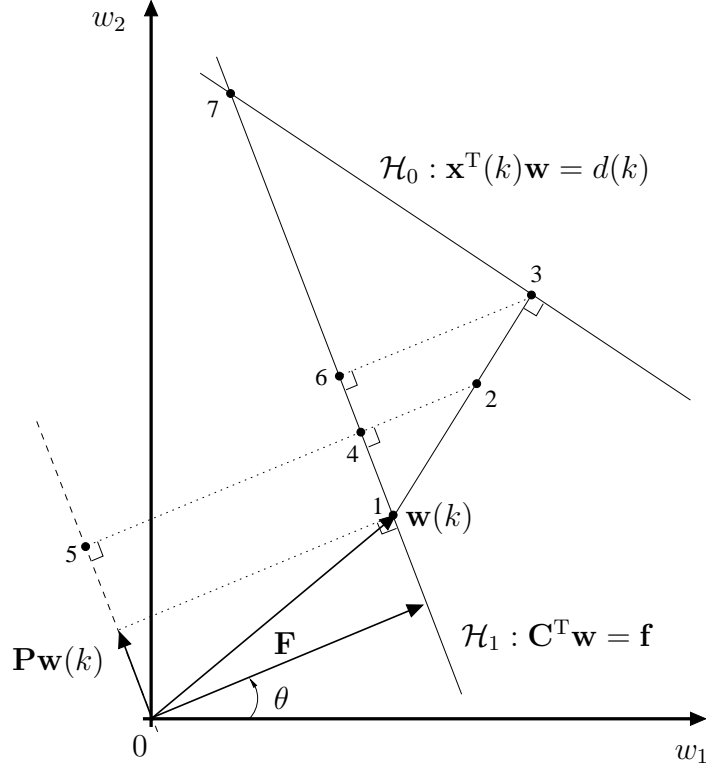


Figure 2.2: Geometrical interpretation of some constrained algorithms.

1. $\mathbf{w}(k) = \mathbf{F} + \mathbf{P}\mathbf{w}(k)$
2. $\mathbf{w}(k+1)$ for the unconstrained LMS algorithm.
3. $\mathbf{w}(k+1)$ for the unconstrained NLMS algorithm.
4. $\mathbf{w}(k+1)$ for the constrained LMS algorithm [27].
5. $\mathbf{P}\mathbf{w}(k+1)$ for the constrained LMS algorithm [27].
6. $\mathbf{w}(k+1)$ for the projected NLMS algorithm [90].
7. $\mathbf{w}(k+1)$ for the constrained NLMS algorithm [86].

where the error vector $\mathbf{e}(k)$ is defined as

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w} \quad (2.14)$$

and

$$\mathbf{d}(k) = [d(k) \ \lambda^{1/2}d(k-1) \ \dots \ \lambda^{k/2}d(0)]^T \quad (2.15)$$

$$\mathbf{X}(k) = [\mathbf{x}(k) \ \lambda^{1/2}\mathbf{x}(k-1) \ \dots \ \lambda^{k/2}\mathbf{x}(0)] \quad (2.16)$$

are the $(k+1) \times 1$ reference vector and the $MN \times (k+1)$ input matrix, respectively, and λ is the forgetting factor ($0 < \lambda \leq 1$). Applying the method of Lagrange multipliers gives

the constrained LS solution at time instant k [91]

$$\mathbf{w}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k) + \mathbf{R}^{-1}(k)\mathbf{C}[\mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{C}]^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{p}(k)] \quad (2.17)$$

where $\mathbf{R}(k)$ is the $MN \times MN$ *deterministic* correlation matrix and $\mathbf{p}(k)$ is the $MN \times 1$ *deterministic* cross-correlation vector defined as

$$\mathbf{R}(k) = \mathbf{X}(k)\mathbf{X}^T(k) = \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i)\mathbf{x}^T(i) \quad (2.18)$$

$$\mathbf{p}(k) = \mathbf{X}(k)\mathbf{d}(k) = \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i)d(i). \quad (2.19)$$

A recursive update of the optimal LS solution in (2.17) will now be addressed. First we note that the solution in (2.17) can be divided into two terms

$$\mathbf{w}(k) = \mathbf{w}_{uc}(k) + \mathbf{w}_c(k) \quad (2.20)$$

where

$$\mathbf{w}_{uc}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k) \quad (2.21)$$

and

$$\mathbf{w}_c(k) = \mathbf{R}^{-1}(k)\mathbf{C}[\mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{C}]^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{p}(k)] \quad (2.22)$$

The coefficient vector $\mathbf{w}_{uc}(k)$ is an unconstrained solution (the deterministic Wiener solution) and is independent of the constraints, whereas $\mathbf{w}_c(k)$ depends on the constraints imposed by $\mathbf{C}^T\mathbf{w}(k) = \mathbf{f}$. The coefficient vector $\mathbf{w}_{uc}(k)$ already has a recursive expression given by the unconstrained RLS algorithm [12]

$$\mathbf{w}_{uc}(k) = \mathbf{w}_{uc}(k-1) + e_{uc}(k)\boldsymbol{\kappa}(k) \quad (2.23)$$

where $\boldsymbol{\kappa}(k) = \mathbf{R}^{-1}(k)\mathbf{x}(k)$ is the gain vector, and $e_{uc}(k) = d(k) - \mathbf{w}_{uc}^T(k-1)\mathbf{x}(k)$ is the *a priori* unconstrained error.

In order to derive a recursive update for $\mathbf{w}_c(k)$, let us define the auxiliary matrices $\mathbf{\Gamma}(k)$ and $\mathbf{\Psi}(k)$, which have dimensions $(MN \times p)$ and $(p \times p)$, respectively.

$$\mathbf{\Gamma}(k) = \mathbf{R}^{-1}(k)\mathbf{C} \quad (2.24)$$

$$\mathbf{\Psi}(k) = \mathbf{C}^T \mathbf{\Gamma}(k) = \mathbf{C}^T \mathbf{R}^{-1}(k) \mathbf{C} \quad (2.25)$$

such that

$$\mathbf{w}_c(k) = \mathbf{\Gamma}(k) \mathbf{\Psi}^{-1}(k) [\mathbf{f} - \mathbf{C}^T \mathbf{w}_{uc}(k)] \quad (2.26)$$

For the case of a single constraint, $\mathbf{\Psi}(k)$ is a scalar and computation of $\mathbf{\Psi}^{-1}(k)$ is trivial. In case of multiple constraints, a recursive formula will be required to reduce the computational complexity. Since we are not concerned here with the derivation of efficient updating schemes, Table 2.1 shows only the basic recursions of the CRLS algorithm as stated by Equations (2.23)–(2.26). For a more efficient implementation of the CRLS algorithm, see [36, 85]. For the equivalence study to be carried out in Section 2.5 we show in Appendix A2.1 of this chapter that the CRLS recursions can be written as

$$\mathbf{w}(k) = \mathbf{w}(k-1) + e(k) \mathbf{R}^{-1}(k) \mathbf{x}(k) - e(k) \mathbf{R}^{-1}(k) \mathbf{C} [\mathbf{C}^T \mathbf{R}^{-1}(k) \mathbf{C}]^{-1} \mathbf{C}^T \mathbf{R}^{-1}(k) \mathbf{x}(k) \quad (2.27)$$

Equation (2.27) is of pure theoretical interest, since it will not render an efficient implementation.

2.4 The Generalized Sidelobe Canceling Model

Many implementations of LC adaptive filters utilize the advantages of the GSC model [12], mainly because this model employs unconstrained adaptation algorithms that have been extensively studied in the literature. Figure 2.3 shows the schematic of the GSC model.

Let \mathbf{B} in Figure 2.3 be a full-rank $MN \times (MN - p)$ blocking matrix designed to filter out completely the components of the input signal that are in the same direction as the constraints. Matrix \mathbf{B} must span the null space of the constraint matrix \mathbf{C} , i.e.,

$$\mathbf{B}^T \mathbf{C} = \mathbf{0} \quad (2.28)$$

In order to relate the GSC model and the linearly constrained minimum variance (LCMV) filter, let \mathbf{T} be an $MN \times MN$ transformation matrix such that

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{B} \end{bmatrix} \quad (2.29)$$

Table 2.1: The Constrained RLS Algorithm.

Initialization:

$\mathbf{w}_{uc}(0)$ and $\mathbf{R}^{-1}(0)$

for each k

{

$\mathbf{k}(k) = \mathbf{R}^{-1}(k-1)\mathbf{x}(k)$

$\boldsymbol{\kappa}(k) = \frac{\mathbf{k}(k)}{\lambda + \mathbf{x}^T(k)\mathbf{k}(k)}$

$\mathbf{R}^{-1}(k) = \frac{1}{\lambda} \left[\mathbf{R}^{-1}(k-1) - \frac{\mathbf{k}(k)\mathbf{k}^T(k)}{\lambda + \mathbf{x}^T(k)\mathbf{k}(k)} \right]$

$e_{uc}(k) = d(k) - \mathbf{w}_{uc}^T(k-1)\mathbf{x}(k)$

$\mathbf{w}_{uc}(k) = \mathbf{w}_{uc}(k-1) + e_{uc}(k)\boldsymbol{\kappa}(k)$

$\boldsymbol{\Gamma}(k) = \mathbf{R}^{-1}(k)\mathbf{C}$

$\boldsymbol{\Psi}(k) = \mathbf{C}^T\boldsymbol{\Gamma}(k)$

$\mathbf{w}(k) = \mathbf{w}_{uc}(k) + \boldsymbol{\Gamma}(k)\boldsymbol{\Psi}^{-1}(k) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k)]$

}

Now suppose that a transformed coefficient vector $\bar{\mathbf{w}}$ relates to the LCMV coefficient vector \mathbf{w} through

$$\mathbf{w} = \mathbf{T}\bar{\mathbf{w}} \quad (2.30)$$

This transformation of the coefficient vector does not modify the output error [69] as long as \mathbf{T} is invertible, which is always guaranteed from (2.28). If we partition vector $\bar{\mathbf{w}}$ as

$$\bar{\mathbf{w}} = \begin{bmatrix} \bar{\mathbf{w}}_U \\ -\bar{\mathbf{w}}_L \end{bmatrix} \quad (2.31)$$

with $\bar{\mathbf{w}}_U$ and $\bar{\mathbf{w}}_L$ vectors of dimension $p \times 1$ and $(MN - p) \times 1$, respectively, it can be shown that we have [12]

$$\mathbf{w} = \mathbf{F} - \mathbf{B}\bar{\mathbf{w}}_L \quad (2.32)$$

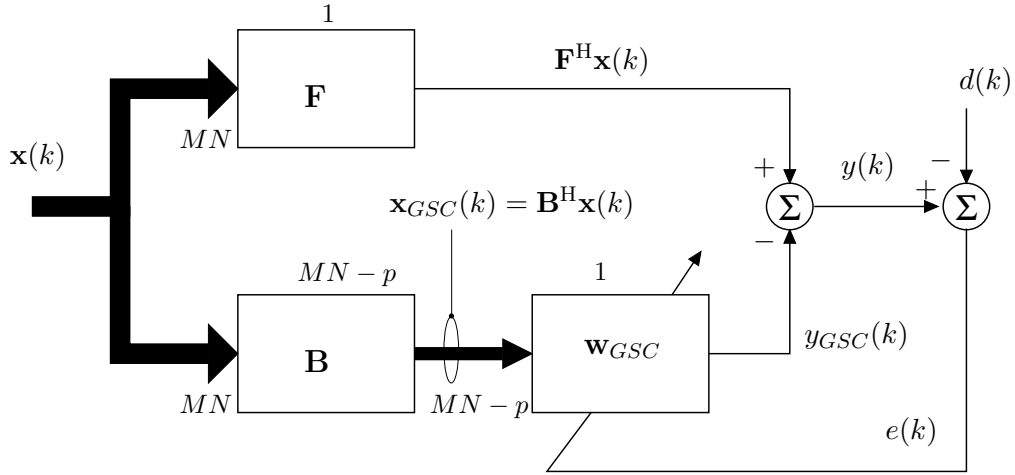


Figure 2.3: Generalized sidelobe canceling (GSC) model.

and

$$\bar{\mathbf{w}}_U = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{f} \quad (2.33)$$

$$\bar{\mathbf{w}}_L = \mathbf{w}_{GSC} \quad (2.34)$$

where $\mathbf{F} = \mathbf{C} \bar{\mathbf{w}}_U$ is the constant part of vector $\bar{\mathbf{w}}$ that satisfies the constraints, i.e., $\mathbf{C}^T \mathbf{w} = \mathbf{C}^T \mathbf{F} = \mathbf{f}$ [12]. Vector \mathbf{w}_{GSC} is not affected by the constraints and may be adapted using unconstrained adaptation algorithms in order to reduce interference from signal components that lie within the null space of \mathbf{C} . For example, the GSC implementation of the LMS algorithm (GSC-LMS) is, using the notation in Figure 2.3, given by

$$\mathbf{w}_{GSC}(k+1) = \mathbf{w}_{GSC}(k) + \mu e_{GSC}(k) \mathbf{x}_{GSC}(k).$$

The desired signal as defined in Figure 2.3, incorporates an external reference signal $d(k)$ such that the resulting desired signal fed back to the adaptation algorithm in the GSC structure becomes $d_{GSC}(k) = \mathbf{F}^T \mathbf{x}(k) - d(k)$. This more general case includes common applications where $d(k) = 0$, e.g., blind beamforming [27] and blind multiuser detection [33].

It is clear from the previous discussion that coefficient adaptation for the GSC model is performed within a reduced-dimension subspace. The transformation \mathbf{T} in (2.29) applied onto the input-signal vector is such that the lower part of this transformed input $\mathbf{B}^T \mathbf{x}(k)$ is restricted to the null space of the constraint matrix \mathbf{C} , which is of dimension $MN - p$.

Therefore adaptation along $\mathbf{B}^T \mathbf{x}(k)$ does not cause any departure from the constraint hyperplane. Another important factor to stress is that the input signal of the adaptive filter is persistently exciting of order $MN - p$.

2.4.1 The GSC-RLS Algorithm

This section provides the details related to the GSC-RLS implementation necessary for the equivalence study in Section 2.5. The optimal least-squares solution of $\mathbf{w}_{GSC}(k)$ is given by

$$\mathbf{w}_{GSC}(k) = \mathbf{R}_{GSC}^{-1}(k) \mathbf{p}_{GSC}(k) \quad (2.35)$$

where $\mathbf{R}_{GSC}(k)$ and $\mathbf{p}_{GSC}(k)$ are the deterministic autocorrelation matrix and cross-correlation vector, respectively. From Figure 2.3 it follows that

$$\mathbf{R}_{GSC}(k) = \sum_{i=0}^k \lambda^{k-i} \mathbf{x}_{GSC}(k) \mathbf{x}_{GSC}^T(k) = \mathbf{B}^T \mathbf{R}(k) \mathbf{B} \quad (2.36)$$

$$\mathbf{p}_{GSC}(k) = \sum_{i=0}^k \lambda^{k-i} [\mathbf{F}^T \mathbf{x}(k) - d(k)] [\mathbf{B}^T \mathbf{x}(k)] = -\mathbf{B}^T \mathbf{p}(k) + \mathbf{B}^T \mathbf{R}(k) \mathbf{F} \quad (2.37)$$

The RLS recursions for the GSC structure becomes

$$\mathbf{w}_{GSC}(k) = \mathbf{w}_{GSC}(k-1) + e_{GSC}(k) \boldsymbol{\kappa}_{GSC}(k) \quad (2.38)$$

where $e_{GSC}(k) = d_{GSC}(k) - \mathbf{w}_{GSC}^T(k-1) \mathbf{x}_{GSC}(k)$ is the *a priori error* and $\boldsymbol{\kappa}_{GSC}(k) = \mathbf{R}_{GSC}^{-1}(k) \mathbf{x}_{GSC}(k)$ is the gain vector. In the next section we will compare the recursions in Equation (2.38) with those of the CRLS algorithm in Equation (2.27).

2.5 Equivalence of LC and GSC Adaptive Filters

This section addresses the relationship between the LC adaptive filters and their corresponding GSC implementation. It is well known that the CLMS and GSC-LMS as well as the CRLS and the GSC-RLS formulations have the same optimal solution [28, 91]. Analysis of the CLMS and GSC-LMS algorithms reveals that the transients of both algorithms only

become equal if \mathbf{B} is orthogonal, i.e., $\mathbf{B}^T \mathbf{B} = \mathbf{I}$ [28]. The requirement of a orthogonal blocking matrix can lead to a computationally complex implementation of the GSC structure. This is because the computations required for the multiplication of the input-signal vector with the blocking matrix may exceed the filtering operation by an order of magnitude. In these situations employing other approaches may be more efficient [66, 67]. For the case of non-orthogonal matrices the transient, or equivalently, the convergence speed, depends on the step size and the particular blocking matrix chosen. In other words, if the blocking matrix changes, the step size changes, including the limits for stability. To our knowledge, no results comparing the transient behavior of the CRLS and the GSC-RLS algorithms has been provided before. Our main goal for the remaining part of this section is to investigate under what circumstances the transients of the CRLS algorithm in Section 2.3.2 and the GSC-RLS algorithm in Section 2.4.1 are identical.

We will study the coefficient-vector evolution defined as

$$\Delta \mathbf{v}(k) = \mathbf{w}(k) - \mathbf{w}(k-1) \quad (2.39)$$

Equation (2.27) gives us the coefficient-vector evolution for the CRLS algorithm as

$$\Delta \mathbf{v}(k) = e(k) \left\{ \mathbf{I} - \mathbf{R}^{-1}(k) \mathbf{C} [\mathbf{C}^T \mathbf{R}^{-1}(k) \mathbf{C}]^{-1} \mathbf{C}^T \right\} \mathbf{R}^{-1}(k) \mathbf{x}(k) \quad (2.40)$$

For the GSC-RLS algorithm, considering that $\mathbf{w}(k) = \mathbf{F} - \mathbf{B} \mathbf{w}_{GSC}(k)$, Equation (2.38) gives us

$$\begin{aligned} \Delta \mathbf{v}(k) &= \mathbf{F} - \mathbf{B} \mathbf{w}_{GSC}(k) - \mathbf{w}(k-1) \\ &= \mathbf{F} - \mathbf{B} [\mathbf{w}_{GSC}(k-1) + e_{GSC}(k) \mathbf{R}_{GSC}^{-1}(k) \mathbf{x}_{GSC}(k)] - \mathbf{w}(k-1) \\ &= e(k) \mathbf{B} [\mathbf{B}^T \mathbf{R}(k) \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{x}(k) \\ &= e(k) \left\{ \mathbf{B} [\mathbf{B}^T \mathbf{R}(k) \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{R}(k) \right\} \mathbf{R}^{-1}(k) \mathbf{x}(k) \end{aligned} \quad (2.41)$$

where $\mathbf{w}(k-1) = \mathbf{F} - \mathbf{B} \mathbf{w}_{GSC}(k-1)$ was used together with Equation (2.36). In order for Equations (2.40) and (2.41) to be identical it is required that the following matrix equality holds:

$$\mathbf{B} [\mathbf{B}^T \mathbf{R}(k) \mathbf{B}]^{-1} \mathbf{B}^T \mathbf{R}(k) + \mathbf{R}^{-1}(k) \mathbf{C} [\mathbf{C}^T \mathbf{R}^{-1}(k) \mathbf{C}]^{-1} \mathbf{C}^T = \mathbf{I} \quad (2.42)$$

The initialization of both schemes (CRLS and GSC-RLS) should be equivalent, which means $\mathbf{R}_{GSC}^{-1}(0) = [\mathbf{B}^T \mathbf{R}(0) \mathbf{B}]^{-1}$ and $\mathbf{w}(0) = \mathbf{F} - \mathbf{B} \mathbf{w}_{GSC}(0)$. The equivalence of the CRLS and the GSC-RLS using the correct initialization is ensured by the following lemma:

Lemma 1. *For $\mathbf{B}^T \mathbf{C} = \mathbf{0}$, if $\mathbf{R}^{-1}(k)$ exists and is symmetric, if $\text{rank}(\mathbf{B}) = MN - p$, and if $\text{rank}(\mathbf{C}) = p$, Equation (2.42) holds true.*

Proof. Define the matrices $\bar{\mathbf{B}} = \mathbf{R}^{T/2}(k) \mathbf{B}$ and $\bar{\mathbf{C}} = \mathbf{R}^{-1/2}(k) \mathbf{C}$ where $\mathbf{R}(k) = \mathbf{R}^{1/2}(k) \mathbf{R}^{T/2}(k)$. With these notations, the left hand side of Equation (2.42) becomes $\bar{\mathbf{B}}(\bar{\mathbf{B}}^T \bar{\mathbf{B}})^{-1} \bar{\mathbf{B}}^T + \bar{\mathbf{C}}(\bar{\mathbf{C}}^T \bar{\mathbf{C}})^{-1} \bar{\mathbf{C}}^T$, and it remains to show that this addition of matrices equals identity. For this purpose, let us introduce the matrix $\bar{\mathbf{T}} = [\bar{\mathbf{C}} \ \bar{\mathbf{B}}]$. $\bar{\mathbf{T}}$ is a full-rank $(MN \times MN)$ matrix, and, consequently, $\bar{\mathbf{T}}^{-1}$ exists. We have,

$$\bar{\mathbf{T}}^T \bar{\mathbf{T}} = \begin{bmatrix} \bar{\mathbf{C}}^T \bar{\mathbf{C}} & \bar{\mathbf{C}}^T \bar{\mathbf{B}} \\ \bar{\mathbf{B}}^T \bar{\mathbf{C}} & \bar{\mathbf{B}}^T \bar{\mathbf{B}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{C}}^T \bar{\mathbf{C}} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{B}}^T \bar{\mathbf{B}} \end{bmatrix} \quad (2.43)$$

where the relation $\bar{\mathbf{B}}^T \bar{\mathbf{C}} = \mathbf{0}$ was used. We have

$$(\bar{\mathbf{T}}^T \bar{\mathbf{T}})^{-1} = \begin{bmatrix} (\bar{\mathbf{C}}^T \bar{\mathbf{C}})^{-1} & \mathbf{0} \\ \mathbf{0} & (\bar{\mathbf{B}}^T \bar{\mathbf{B}})^{-1} \end{bmatrix} \quad (2.44)$$

Therefore,

$$\begin{aligned} \bar{\mathbf{T}}(\bar{\mathbf{T}}^T \bar{\mathbf{T}})^{-1} \bar{\mathbf{T}}^T &= \bar{\mathbf{T}} \bar{\mathbf{T}}^{-1} \bar{\mathbf{T}}^{-T} \bar{\mathbf{T}}^T = \mathbf{I} \\ &= [\bar{\mathbf{C}} \ \bar{\mathbf{B}}] \begin{bmatrix} (\bar{\mathbf{C}}^T \bar{\mathbf{C}})^{-1} & \mathbf{0} \\ \mathbf{0} & (\bar{\mathbf{B}}^T \bar{\mathbf{B}})^{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{C}}^T \\ \bar{\mathbf{B}}^T \end{bmatrix} \\ &= \bar{\mathbf{C}}(\bar{\mathbf{C}}^T \bar{\mathbf{C}})^{-1} \bar{\mathbf{C}}^T + \bar{\mathbf{B}}(\bar{\mathbf{B}}^T \bar{\mathbf{B}})^{-1} \bar{\mathbf{B}}^T = \mathbf{I} \end{aligned} \quad (2.45)$$

■

As a consequence of Lemma 1, and Equations (2.40) and (2.41), we can conclude that the necessary requirement for equivalent transients of the CRLS and the GSC-RLS algorithms is that $\mathbf{B}^T \mathbf{C} = \mathbf{0}$, which holds true in any GSC structure. This is a looser requirement than the transient-equivalence of the CLMS and GSC-LMS algorithms, which in addition to $\mathbf{B}^T \mathbf{C} = \mathbf{0}$, requires \mathbf{B} to be orthogonal.

2.6 Choice of Blocking Matrix \mathbf{B}

The structure of matrix \mathbf{B} plays an important role in the GSC structure, for its choice determines the computational complexity and, in many cases, the robustness against numerical instabilities of the overall system [33]. If singular value decomposition or any other decomposition is employed, the resulting non-squared $(MN \times MN - p)$ matrix \mathbf{B} will, in general, have no exploitable special structure. This may result in a highly inefficient implementation with computational complexity up to one order of magnitude higher than that of the adaptation algorithm itself. This is due to the matrix-vector multiplication $\mathbf{B}\mathbf{x}(k)$ performed at each iteration.

A non-orthogonal blocking matrix suggested in [92] is implemented as a sequence of sparse blocking matrices $\mathbf{B} = \mathbf{B}_1 \cdots \mathbf{B}_{p-1} \mathbf{B}_p$ where \mathbf{B}_i is an $(MN - i + 1) \times (MN - i)$ matrix of full rank. A straightforward choice of \mathbf{B}_i is

$$\mathbf{B}_i^T = \begin{bmatrix} \tilde{c}_{i,2} & -\tilde{c}_{i,1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \tilde{c}_{i,MN-i+1} & -\tilde{c}_{i,MN-i} \end{bmatrix} \quad (2.46)$$

where $\tilde{c}_{i,j}$ denotes the (i, j) th element the matrix $\tilde{\mathbf{C}}_{i-1} = \mathbf{B}_{i-1}^T \mathbf{B}_{i-2}^T \cdots \mathbf{B}_1^T \mathbf{C}$. To illustrate the procedure discussed above, consider the simplified example below.

EXAMPLE 2.1

Given the constraint matrix

$$\mathbf{C} = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{bmatrix},$$

construct a blocking matrix as a sequence of sparse blocking matrices using Equation (2.46).

SOLUTION

In this example, the number of constraints equals $p = 2$ and, consequently, the blocking matrix can be constructed as a sequence of two blocking matrices $\mathbf{B} = \mathbf{B}_1 \mathbf{B}_2$. The first blocking matrix \mathbf{B}_1 is designed to null out the first column of \mathbf{C} and, therefore, we have

$\tilde{c}_{i,j} = c_{i,j}$. Using (2.46) gives us

$$\mathbf{B}_1^T = \begin{bmatrix} c_{1,2} & -c_{1,1} & 0 \\ 0 & c_{1,3} & -c_{1,2} \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 \\ 0 & 3 & -2 \end{bmatrix} \quad (2.47)$$

The matrix $\tilde{\mathbf{C}}_1$ becomes

$$\tilde{\mathbf{C}}_1 = \mathbf{B}_1^T \mathbf{C} = \begin{bmatrix} 0 & 4 \\ 0 & 4 \end{bmatrix}$$

and the second blocking matrix is now easily constructed as

$$\mathbf{B}_2^T = [\tilde{c}_{2,2} \quad -\tilde{c}_{2,1}] = [4 \quad -4]$$

△

A similar blocking matrix to the one discussed above with slightly lower complexity was presented in [69] for the case of $p = 1$ constraint. The simple blocking matrix above reduces the overall complexity considerably but will, for example, not be directly applicable to the problem of LCMV filtering of sinusoids that was considered in [85]. In beamforming with presteering the requirement for spatial blocking of the look direction is that the rows of \mathbf{B} sum up to zero [28]. A commonly used blocking matrix fulfilling this requirement contains 1 and -1 along the diagonals, and is obtained by Equation (2.46) using $\tilde{c}_j = 1$. If the number of antennas is such that $M = 2^L$, $L = \mathbb{Z}_+$, an orthogonal blocking matrix can be constructed easily using Walsh functions [28].

Although in some applications [28] it may be possible to construct trivial blocking matrices whose elements are either 0, 1, or -1 , these matrices pose some practical problems that may prevent their use. For instance, if matrix \mathbf{B} is such that the transformation matrix \mathbf{T} is not orthogonal, then the transients of the adaptive filters in the GSC model and in the LCMV may bear no relation [28]. Furthermore, if applied to the multistage Wiener filter structure presented in [69], non-orthogonal transformations invariably yield severe problems related to finite precision arithmetic [33].

The Householder decomposition as suggested in Chapter 3 allows efficient implementation and results in a orthogonal transformation matrix. If necessary, the Householder reflections can be performed via dedicated CORDIC hardware or software [93].

2.7 Simulation Results

In this section the equivalence of the CRLS and GSC-RLS algorithms is investigated in two applications. The first application is a beamforming application where the desired signal is set to zero, i.e., $d(k) = 0$. The second application using the more general desired signal with $d(k) \neq 0$ is a system-identification application where the adaptive filter is constrained to have linear phase.

2.7.1 Beamforming with Derivative Constraints

A uniform linear array with $M = 12$ antennas with element spacing equal to half wavelength was used in a system with $K = 5$ users, where the signal of one user is of interest and the other 4 are treated as interferers. The system is assumed to be narrowband and, therefore, only one filter coefficient per antenna, $N = 1$, is necessary. The received discrete-time signal can be written as

$$\mathbf{x}(k) = \mathbf{S}\mathbf{A}\mathbf{u}(k) + \mathbf{n}(k)$$

where $\mathbf{S} = [\mathbf{s}(\theta_1) \ \mathbf{s}(\theta_2) \ \dots \ \mathbf{s}(\theta_K)]$ is the *steering matrix* containing the steering vectors of the users given by [37] $\mathbf{s}(\theta_i) = [e^{j(1-m_0)\pi \sin \theta_i} \ e^{j(2-m_0)\pi \sin \theta_i} \ \dots \ e^{j(M-m_0)\pi \sin \theta_i}]^T$, θ_i being the direction of arrival (DOA), $\mathbf{A} = \text{diag}[A_1 \ A_2 \ \dots \ A_K]$ contains the user amplitudes A_i , $\mathbf{u}(k) = [u_1(k) \ u_2(k) \ \dots \ u_K(k)]^T$ is a vector of the transmitted user information, and $\mathbf{n}(k)$ is the sampled noise sequence. The parameter m_0 specifies a reference antenna which is used as phase reference, here set to $m_0 = 3$. The direction of arrival (DOA) and the signal-to-noise ratio (SNR) for the different signals can be found in Table 2.2. A second-order derivative constraint matrix [37] was used giving a total of three constraints (see [37] for further details) and the constraint matrix

$$\mathbf{C} = [\mathbf{c}_0 \ \mathbf{c}_1 \ \mathbf{c}_2]$$

where $\mathbf{c}_i = [(1 - m_0)^i \ (2 - m_0)^i \ \dots \ (M - m_0)^i]^T$. We used a non-orthogonal blocking matrix that was constructed through a sequence of sparse matrices as proposed in [92],

Table 2.2: Signal Parameters

| SIGNAL | DOA θ_i | SNR |
|--------------|----------------|-------|
| desired | 0° | 15 dB |
| interferer 1 | 22° | 20 dB |
| interferer 2 | -15° | 25 dB |
| interferer 3 | -20° | 25 dB |
| interferer 4 | -50° | 20 dB |

which was also reviewed in Section 2.6, rendering an implementation of the multiplication $\mathbf{B}\mathbf{x}(k)$ of low computational complexity.

The CRLS and the GSC-RLS algorithms used $\lambda = 0.99$. Figure 2.4 shows the evolution of coefficient-error norm for the CRLS and the GSC-RLS algorithms. Figure 2.4 also plots the results for the CLMS and the GSC-LMS algorithms. As can be seen from the figure, the CLMS and the GSC-LMS algorithms only become identical when using the orthogonal blocking matrix, whereas the CRLS and the GSC-RLS algorithms are identical for the non-orthogonal blocking matrix. This fact is further illustrated in Figure 2.5, where the norm of the difference between the CRLS and the GSC-RLS solutions is plotted.

2.7.2 Identification of Plant with Linear Phase

An experiment was carried out in a system-identification problem where the filter coefficients were constrained to preserve linear phase at every iteration. For this example we made $N = 11$ and, in order to fulfill the linear phase requirement, we made

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{(N-1)/2} \\ \mathbf{0}^T \\ -\mathbf{J}_{(N-1)/2} \end{bmatrix} \quad (2.48)$$

with \mathbf{J} being a reversal matrix (an identity matrix with all rows in reversed order), and

$$\mathbf{f} = [0 \cdots 0]^T \quad (2.49)$$

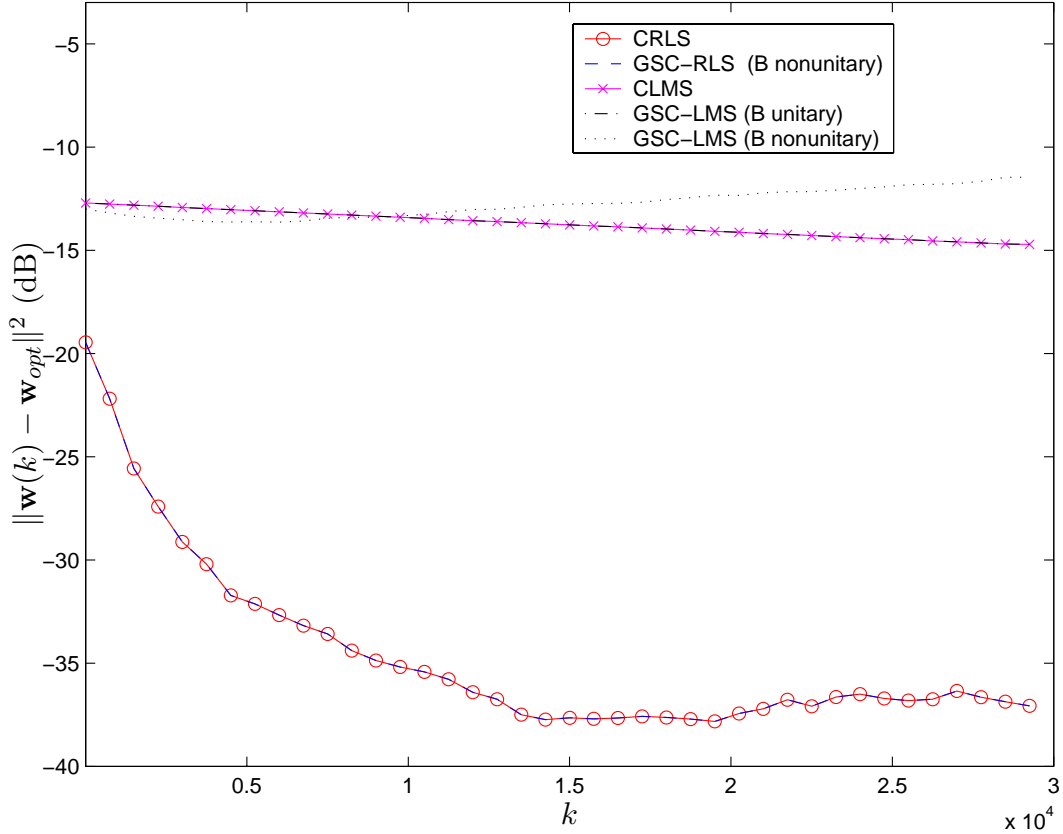


Figure 2.4: Coefficient-error vector as a function of the iteration k for a beamforming application using derivative constraints.

Due to the symmetry of \mathbf{C} and the fact that \mathbf{f} is a null vector, more efficient structures can be employed [36]. We used the non-orthogonal blocking matrix given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{0} \\ \mathbf{0}^T & 1 \\ \mathbf{J}_{N/2} & \mathbf{0} \end{bmatrix} \quad (2.50)$$

The input signal consists of zero-mean unity-variance colored noise with eigenvalue spread around 1350 and the reference signal was obtained after filtering the input by a linear-phase FIR filter and adding measurement noise with variance equal to 10^{-6} .

The CRLS and the GSC-RLS algorithms used $\lambda = 0.95$. Figure 2.6 shows the evolution of coefficient-error norm for the CRLS and the GSC-RLS algorithms. Similarly as in the beamforming example, the curves for the CRLS and the GSC-RLS algorithms are identical, and the CLMS and the GSC-LMS become identical only when the blocking

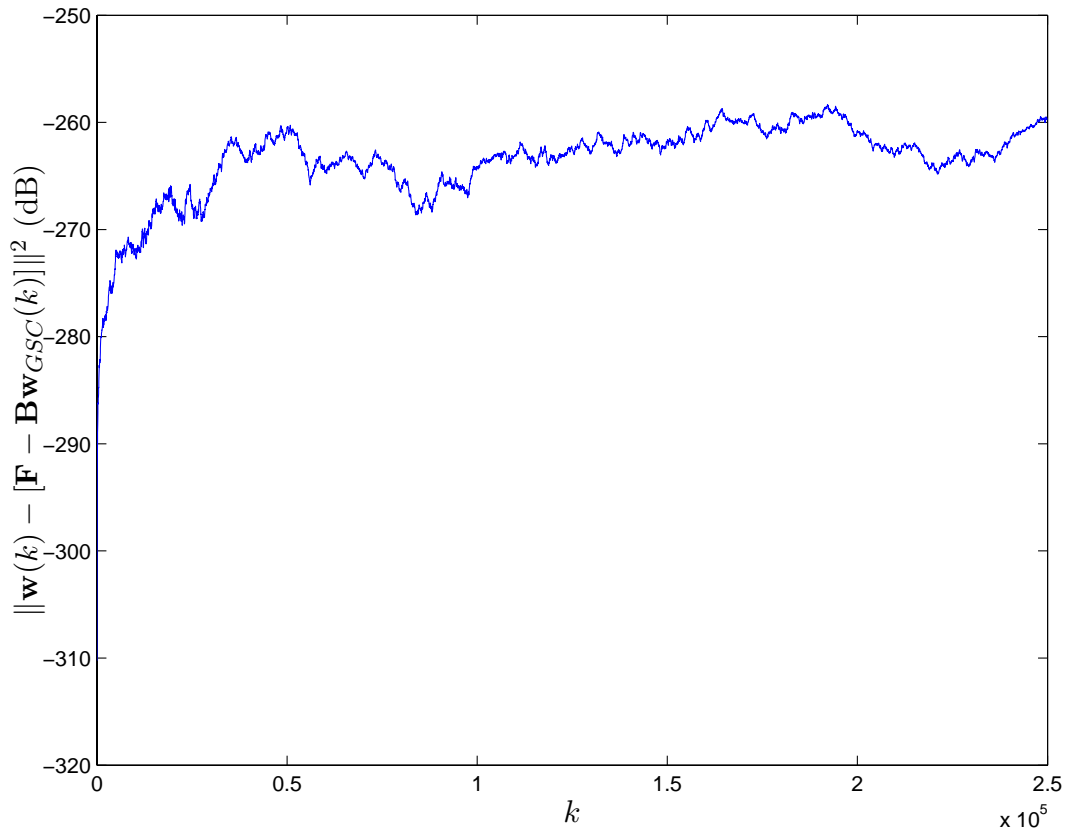


Figure 2.5: Norm of the difference of the CRLS and the GSC-RLS coefficient vectors as a function of the iteration k for a beamforming application using derivative constraints.

matrix is orthogonal. Figure 2.7, plots the norm of the difference between the CRLS and the GSC-RLS solutions.

2.8 Conclusions

This section reviewed the area of constrained adaptive filters and presented theoretical results linking transient behavior of the constrained RLS algorithm and the GSC structure with the RLS algorithm. We showed that, contrary to the LMS algorithm, in the case of the RLS algorithm transient behavior can always be ensured to be identical in the two forms of implementation provided only that the blocking matrix and the constraint matrix span orthogonal subspaces. This result facilitates algorithm tuning, because it establishes that the constrained algorithm behaves exactly like its unconstrained counterpart in transient

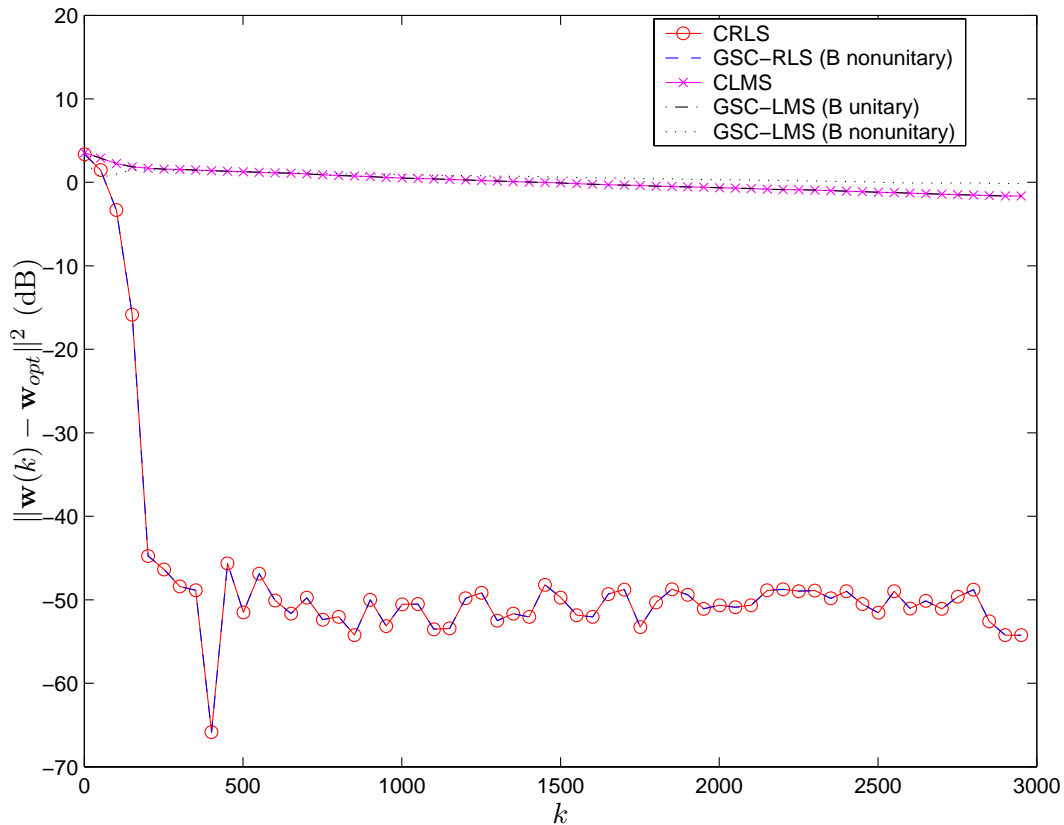


Figure 2.6: Coefficient-error vector as a function of the iteration k for a system-identification application.

as well as in steady state. This confirms intuition, because both implementations solve the same LS problem exactly. The result presented here may favor the utilization of the unconstrained counterpart of the CRLS algorithm, because it facilitates the choice of various versions of the RLS algorithm optimized with respect to computational complexity and robustness.

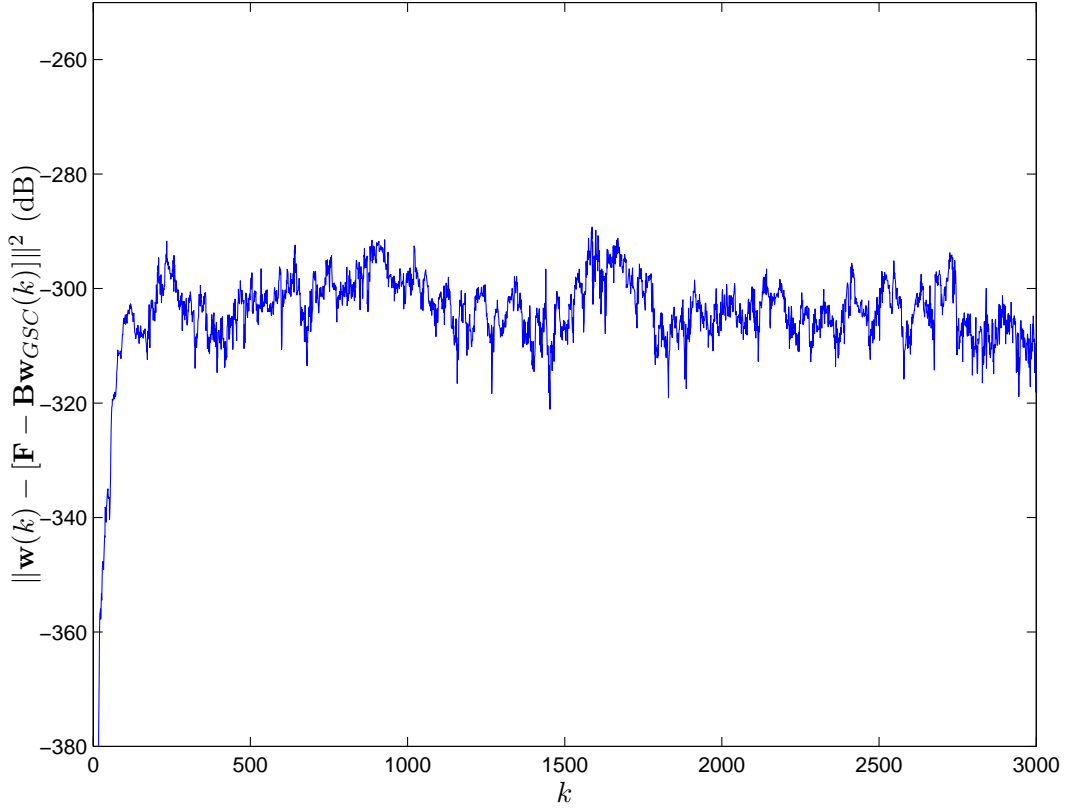


Figure 2.7: Norm of the difference of the CRLS and the GSC-RLS coefficient vectors as a function of the iteration k for a system-identification application.

Appendix A2.1

In this Appendix it is shown that the CRLS algorithm [85] can be written on the form given by Equation (2.27). Let us start by finding a recursive expression of $\mathbf{\Gamma}(k)$. We first note that $\mathbf{R}^{-1}(k)$ can be written as [12]

$$\mathbf{R}^{-1}(k) = \frac{1}{\lambda} [\mathbf{R}^{-1}(k-1) - \mathbf{R}^{-1}(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1)] \quad (2.51)$$

Using (2.51) in (2.24) results

$$\begin{aligned} \mathbf{\Gamma}(k) &= \frac{1}{\lambda} [\mathbf{R}^{-1}(k-1) \mathbf{C} - \mathbf{R}^{-1}(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{R}^{-1}(k-1) \mathbf{C}] \\ &= \frac{1}{\lambda} [\mathbf{\Gamma}(k-1) - \mathbf{\kappa}(k) \mathbf{x}^T(k) \mathbf{\Gamma}(k-1)] \end{aligned} \quad (2.52)$$

In order to obtain a recursive expression for $\mathbf{\Psi}^{-1}(k)$, pre-multiply (2.52) by \mathbf{C}^T and

apply the *Matrix Inversion Lemma*

$$\begin{aligned}
\Psi^{-1}(k) &= \lambda \{ \mathbf{C}^T \Gamma(k-1) - \mathbf{C}^T \boldsymbol{\kappa}(k) \mathbf{x}^T(k) \Gamma(k-1) \}^{-1} \\
&= \lambda [\mathbf{A} + \mathbf{BCD}]^{-1} \\
&= \lambda \left\{ \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} [\mathbf{D} \mathbf{A}^{-1} \mathbf{B} + \mathbf{C}^{-1}]^{-1} \mathbf{D} \mathbf{A}^{-1} \right\} \\
&= \lambda \left[\Psi^{-1}(k-1) + \frac{\Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k) \mathbf{x}^T(k) \Gamma(k-1) \Psi^{-1}(k-1)}{1 - \mathbf{x}^T(k) \Gamma(k-1) \Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k)} \right] \quad (2.53)
\end{aligned}$$

In order to simplify the notation, define $\boldsymbol{\ell}(k)$ as

$$\boldsymbol{\ell}(k) = \frac{\Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k)}{1 - \mathbf{x}^T(k) \Gamma(k-1) \Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k)} \quad (2.54)$$

which gives

$$\Psi^{-1}(k) = \lambda [\Psi^{-1}(k-1) + \boldsymbol{\ell}(k) \mathbf{x}^T(k) \Gamma(k-1) \Psi^{-1}(k-1)] \quad (2.55)$$

From (2.54), we know that

$$\boldsymbol{\ell}(k) = \boldsymbol{\ell}(k) \mathbf{x}^T(k) \Gamma(k-1) \Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k) + \Psi^{-1}(k-1) \mathbf{C}^T \boldsymbol{\kappa}(k) \quad (2.56)$$

Post-multiplying (2.53) by $\mathbf{C}^T \boldsymbol{\kappa}(k)$ and dividing by λ gives the same expression as in (2.56), therefore

$$\boldsymbol{\ell}(k) = \frac{1}{\lambda} \Psi^{-1}(k) \mathbf{C}^T \boldsymbol{\kappa}(k) \quad (2.57)$$

To show the formulation of the CRLS algorithm given by (2.27), substitute $e_{uc}(k)$ by

$$e_{uc}(k) = d(k) - \mathbf{w}_I^T(k-1) \mathbf{x}(k),$$

and $\Psi(k)$ and $\Gamma(k)$ by their recursive expressions. Using the recursions given by Equations

tions (2.20) and (2.22), the coefficient update for the CRLS algorithm is given by

$$\begin{aligned}
\mathbf{w}(k) &= \mathbf{w}_{uc}(k) + \mathbf{w}_c(k) \\
&= \mathbf{w}_{uc}(k-1) + \mathbf{\Gamma}(k)\mathbf{\Psi}^{-1}(k)[\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] + e_{uc}(k)\boldsymbol{\kappa}(k) \\
&\quad - \mathbf{\Gamma}(k)\mathbf{\Psi}^{-1}(k)\mathbf{C}^T\boldsymbol{\kappa}(k)e_{uc}(k) \\
&= \mathbf{w}_{uc}(k-1) + \mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \boldsymbol{\kappa}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] + e_{uc}(k)\boldsymbol{\kappa}(k) \\
&\quad + \mathbf{\Gamma}(k-1)\boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \boldsymbol{\kappa}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) \times [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \mathbf{\Gamma}(k)\mathbf{\Psi}^{-1}(k)\mathbf{C}^T\boldsymbol{\kappa}(k)e_{uc}(k) \tag{2.58}
\end{aligned}$$

In Equation (2.58) the first row simplifies to

$$\mathbf{w}_{uc}(k-1) + \mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] = \mathbf{w}(k-1).$$

and the second row using $e_I(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}_{uc}(k-1)$, becomes

$$\begin{aligned}
&\boldsymbol{\kappa}(k) (d(k) - \mathbf{x}^T(k)\{\mathbf{w}_{uc}(k-1) + \mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) \times [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)]\}) \\
&= \boldsymbol{\kappa}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1)] = \boldsymbol{\kappa}(k)e(k).
\end{aligned}$$

As a consequence Equation (2.58) simplifies to

$$\begin{aligned}
\mathbf{w}(k) &= \mathbf{w}(k-1) + \boldsymbol{\kappa}(k)e(k) + \mathbf{\Gamma}(k-1)\boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \boldsymbol{\kappa}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \mathbf{\Gamma}(k) \underbrace{\mathbf{\Psi}^{-1}(k)\mathbf{C}^T\boldsymbol{\kappa}(k)}_{\lambda\boldsymbol{\ell}(k)} [d(k) - \mathbf{x}^T(k)\mathbf{w}_{uc}(k-1)] \\
&= \mathbf{w}(k-1) + \boldsymbol{\kappa}(k)e(k) \\
&\quad + \underbrace{[\mathbf{\Gamma}(k-1) - \boldsymbol{\kappa}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)]}_{\lambda\mathbf{\Gamma}(k)} \boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \mathbf{\Gamma}(k)\lambda\boldsymbol{\ell}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}_{uc}(k-1)] \\
&= \mathbf{w}(k-1) + \boldsymbol{\kappa}(k)e(k) + \lambda\mathbf{\Gamma}(k)\boldsymbol{\ell}(k)\mathbf{x}^T(k)\mathbf{\Gamma}(k-1)\mathbf{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)] \\
&\quad - \lambda\mathbf{\Gamma}(k)\boldsymbol{\ell}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}_{uc}(k-1)]
\end{aligned}$$

$$\begin{aligned}
&= \mathbf{w}(k-1) + \boldsymbol{\kappa}(k)e(k) - \lambda\boldsymbol{\Gamma}(k)\boldsymbol{\ell}(k) \\
&\quad \times (d(k) - \mathbf{x}^T(k) \underbrace{\{\mathbf{w}_{uc}(k-1) + \boldsymbol{\Gamma}(k-1)\boldsymbol{\Psi}^{-1}(k-1) [\mathbf{f} - \mathbf{C}^T\mathbf{w}_{uc}(k-1)]\}}_{\mathbf{w}(k-1)}) \\
&= \mathbf{w}(k-1) + \boldsymbol{\kappa}(k)e(k) - \lambda\boldsymbol{\Gamma}(k)\boldsymbol{\ell}(k) [d(k) - \mathbf{w}^T(k-1)\mathbf{x}(k)] \\
&= \mathbf{w}(k-1) + e(k)\boldsymbol{\kappa}(k) - \lambda e(k)\boldsymbol{\Gamma}(k)\boldsymbol{\ell}(k) \\
&= \mathbf{w}(k-1) + e(k)\mathbf{R}^{-1}(k)\mathbf{x}(k) - e(k)\mathbf{R}^{-1}(k)\mathbf{C}[\mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{C}]^{-1}\mathbf{C}^T\mathbf{R}^{-1}(k)\mathbf{x}(k) \quad (2.59)
\end{aligned}$$

Chapter 3

Householder Constrained Adaptation Algorithms

This chapter introduces and analyzes an efficient and robust implementation of linearly-constrained adaptive filters that utilize Householder transformation (HT). The method allows direct application of any unconstrained adaptation algorithm as in a generalized sidelobe canceler (GSC), but unlike the GSC the HT-based approach always renders efficient and robust implementations. A complete and detailed comparison with the GSC model and a thorough discussion of the advantages of the HT-based approach are also given. Simulations are run in a beamforming application where a linear array of 12 sensors is used. It is verified that not only the HT approach yields efficient and robust implementation of constrained adaptive filters, but also the beampatterns achieved with this method are much closer to the optimal solution than the beampatterns obtained with GSC models of similar computational complexity.

3.1 Introduction

The linearly-constrained adaptation algorithms discussed in Chapter 2 have in common that the direction of update is premultiplied with a rank-deficient projection matrix, which renders them not optimal in the sense of computational complexity. Furthermore, these algorithms have in common a correction factor to ensure the constraints at every iteration.

If this correction factor is not applied, coefficient divergence may occur due to roundoff errors in certain direction that cannot be suppressed [28]. The GSC model, also discussed in Chapter 2, properly reduces the dimension of the coefficient update using a blocking matrix. The structure of the blocking matrix has a direct effect on the overall computational complexity through its multiplication to the input-signal vector at each iteration. Construction of the blocking matrix using, e.g., singular-value decomposition (SVD), results in a matrix with no special structure, which in the GSC model renders high computational complexity per iteration. In these cases, the practical use of the GSC structure is questionable. The extra computations resulting from the product of the blocking matrix by the input-signal vector may exceed those of the adaptation algorithm and filtering operation by up to one order of magnitude. Other types of blocking matrices with sparse structures may be of more practical use from the perspective of computational complexity. Many times such solutions are application dependent and the resulting matrix is, in general, not orthogonal; in these cases, adaptive implementations of the GSC and linearly-constrained minimum-variance (LCMV) filters may bear no relation [28].

The main contributions of this chapter consider efficient implementations of LC adaptive filtering algorithms that overcome the problem of added computational complexity that may occur in the GSC structure. By suitably transforming the input-signal vector using successive Householder transformations, the algorithms may operate on a reduced-dimension subspace and, therefore, does not require updating of all its coefficients. No correction terms need to be applied and the solution satisfies the constraints exactly in each iteration. In addition, as in the GSC structure, the proposed method may be used with any unconstrained adaptive filtering algorithm. A geometrical interpretation is used to illustrate better the use of the Householder transformation. A detailed explanation of the matrices involved in the process is presented and pseudo-code routines are provided. Even in cases where the GSC structure is equivalent to the Householder implementation introduced here, the latter is more efficient.

The organization of the chapter is as follows. In Section 3.2, the new Householder-transform constrained algorithms are presented as a lower-complexity solution for reducing

the subspace in which adaptive-filter coefficients are updated. Relations to the GSC model are made, resulting in a framework where any unconstrained adaptation algorithm can be applied to linearly constrained problems using the proposed method. Section 3.3 contains simulation results, followed by conclusions in Section 3.4.

3.2 Householder-Transform Constrained Algorithms

For a general constrained minimization problem, the multiplication of the blocking matrix by the input-signal vector in a GSC structure may be computationally intensive and, for many applications, not practical. In this section we propose an elegant solution to this problem. The derivation of the first algorithm presented in this section starts from the CLMS algorithm, Equation (2.9), and performs a rotation on vector $\mathbf{P}\mathbf{x}(k)$ in order to make sure that the coefficient vector is never perturbed in a direction not excited by $\mathbf{P}\mathbf{x}(k)$. This can be done if an orthogonal rotation matrix \mathbf{Q} is used as a transformation that will generate a modified coefficient vector $\bar{\mathbf{w}}(k)$ that relates to $\mathbf{w}(k)$ according to

$$\bar{\mathbf{w}}(k) = \mathbf{Q}\mathbf{w}(k) \quad (3.1)$$

We can visualize this operation in Figure 2.2 on page 28 if we imagine axis \mathbf{w}_1 and \mathbf{w}_2 rotated counterclockwise by an angle θ .

If we choose the matrix \mathbf{Q} such that $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ and

$$\bar{\mathbf{C}}(\bar{\mathbf{C}}^T\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^T = \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (3.2)$$

then $\bar{\mathbf{C}} = \mathbf{Q}\mathbf{C}$ satisfies $\mathbf{f} = \bar{\mathbf{C}}^T\bar{\mathbf{w}}(k+1)$ and the transformed projection matrix is such that

$$\begin{aligned} \bar{\mathbf{P}} &= \mathbf{Q}\mathbf{P}\mathbf{Q}^T \\ &= \mathbf{I} - \bar{\mathbf{C}}(\bar{\mathbf{C}}^T\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^T \\ &= \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \end{aligned} \quad (3.3)$$

If $\bar{\mathbf{w}}(0)$ is initialized as

$$\bar{\mathbf{w}}(0) = \bar{\mathbf{C}}(\bar{\mathbf{C}}^T\bar{\mathbf{C}})^{-1}\mathbf{f} = \mathbf{Q}\mathbf{F} \quad (3.4)$$

then its first p elements, $\bar{\mathbf{w}}_U(0)$, need not be updated. The update equation of the proposed algorithm, named the Householder-Transform Constrained LMS [66], is obtained by pre-multiplying (2.9) by \mathbf{Q} :

$$\begin{aligned}
\bar{\mathbf{w}}(k+1) &= \mathbf{Q}\mathbf{w}(k+1) = \mathbf{Q}\{\mathbf{P}[\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)] + \mathbf{F}\} \\
&= [\mathbf{Q}\mathbf{P}\mathbf{Q}^T][\mathbf{Q}\mathbf{w}(k)] + \mu e(k)[\mathbf{Q}\mathbf{P}\mathbf{Q}^T][\mathbf{Q}\mathbf{x}(k)] + \mathbf{Q}\mathbf{F} \\
&= \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{w}}(k) + \mu e(k) \begin{bmatrix} \mathbf{0}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \bar{\mathbf{x}}(k) + \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \mathbf{0} \end{bmatrix} \\
&= \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k+1) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k) \end{bmatrix} + \mu e(k) \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{x}}_L(k) \end{bmatrix} \tag{3.5}
\end{aligned}$$

where $\bar{\mathbf{w}}_L(k)$ and $\bar{\mathbf{x}}_L(k)$ denote the $MN - p$ last elements of vectors $\bar{\mathbf{w}}(k)$ and $\bar{\mathbf{x}}(k)$, respectively. Note that vector $\bar{\mathbf{C}}(\bar{\mathbf{C}}^T\bar{\mathbf{C}})^{-1}\mathbf{f}$ has only p nonzero elements.

Although the solution $\bar{\mathbf{w}}(k)$ is biased by a transformation \mathbf{Q} , the output signal and, consequently, the output error is not modified by the transformation. We conclude, therefore, that the proposed algorithm minimizes the same objective function as the CLMS algorithm.

3.2.1 Choice of the Transformation Matrix \mathbf{Q}

We maintain that matrix \mathbf{Q} in (3.1) may be constructed with successive Householder transformations [94] applied onto each of the p columns of matrix $\mathbf{C}\mathbf{L}$, where \mathbf{L} is the square-root factor of matrix $(\mathbf{C}^T\mathbf{C})^{-1}$, i.e., $\mathbf{L}\mathbf{L}^T = (\mathbf{C}^T\mathbf{C})^{-1}$.

Theorem 1. *If*

$$\mathbf{Q} = \mathbf{Q}_p \cdots \mathbf{Q}_2 \mathbf{Q}_1 \tag{3.6}$$

where

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I}_{i-1 \times i-1} & \mathbf{0}^T \\ \mathbf{0} & \bar{\mathbf{Q}}_i \end{bmatrix} \tag{3.7}$$

and $\bar{\mathbf{Q}}_i$ is an $(MN - i + 1) \times (MN - i + 1)$ Householder transformation matrix on the form $\bar{\mathbf{Q}}_i = \mathbf{I} - 2\bar{\mathbf{v}}_i\bar{\mathbf{v}}_i^T$ [94], then (3.2) is satisfied.

Proof. After $i - 1$ transformations, matrix $\mathbf{Q}_{i-1} \cdots \mathbf{Q}_1 \mathbf{CL}$ may be partitioned as

$$\mathbf{Q}_{i-1} \cdots \mathbf{Q}_1 \mathbf{CL} = \left[\begin{array}{cc} \overbrace{\mathbf{D}^{(i)} \quad \cdots \quad \mathbf{E}^{(i)}}^{i-1} & \overbrace{\mathbf{0} \quad \cdots \quad \mathbf{A}^{(i)}}^{MN-i+1} \\ \cdots & \cdots \\ \mathbf{0} & \cdots \end{array} \right] \left. \begin{array}{l} \}^{i-1} \\ \}^{MN-i+1} \end{array} \right. \quad (3.8)$$

The $i - 1$ Householder transformations make matrix $\mathbf{D}^{(i)}$ upper triangular. Now let $\mathbf{a}_j^{(i)}$ denote the j th column of the $\mathbf{A}^{(i)}$ matrix. It can be shown by carrying out the Householder transformation $\bar{\mathbf{Q}}_i$ on $\mathbf{A}^{(i)}$ that if the columns, viz $\{\mathbf{a}_j^{(i)}, j = 1, \dots, MN - i + 1\}$, of $\mathbf{A}^{(i)}$ satisfy

$$\|\mathbf{a}_j^{(i)}\| = 1 \quad (3.9)$$

$$(\mathbf{a}_i^{(i)})^T \mathbf{a}_j^{(i)} = \delta_{ij} \quad (3.10)$$

then

$$\bar{\mathbf{Q}}_i \mathbf{A}^{(i)} = \begin{bmatrix} \pm 1 & 0 & \cdots & 0 \\ 0 & \star & \cdots & \star \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \star & \cdots & \star \end{bmatrix} \quad (3.11)$$

For $i = 1$, $\bar{\mathbf{Q}}_1 = \mathbf{Q}_1$, $\mathbf{A}^{(1)} = \mathbf{CL}$. Matrix \mathbf{CL} has orthonormal columns, because $(\mathbf{CL})^T \mathbf{CL} = \mathbf{L}^T (\mathbf{LL}^T)^{-1} \mathbf{L} = \mathbf{I}$. Therefore, (3.9) and (3.10) are directly satisfied. By the *fundamental theorem of inner-product invariance in Householder transforms* [95], orthonormality is maintained for $\mathbf{Q}_1 \mathbf{CL}$ and, by induction, (3.9) and (3.10) are also satisfied for any $i > 1$. As a consequence, $\mathbf{D}^{(i)}$ is a diagonal matrix with ± 1 entries and $\mathbf{E}^{(i)}$ is a matrix of zeros. This concludes the proof. ■

Notice that the ± 1 entries in matrix $\mathbf{D}^{(i)}$ result from the robust implementation of the Householder transformation given in [94].

From (3.5) we verify that the algorithm updates the coefficients in a subspace with reduced dimension. The components of vector $\mathbf{w}(k)$ which lie in the subspace of the constraints need not be updated. Due to the equivalence of Householder reflections and Givens rotations [96], a succession of Givens rotations could also be used. However, rotations are

not as efficiently implemented as reflections and computational complexity might render the resulting algorithm not practical.

3.2.2 The Normalized HCLMS Algorithm

A normalized version of the HCLMS algorithm, namely the NHCLMS algorithm [66], can be derived and its update equation is

$$\bar{\mathbf{w}}(k+1) = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k+1) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k) \end{bmatrix} + \mu \frac{e(k)}{\bar{\mathbf{x}}_L^T(k)\bar{\mathbf{x}}_L(k)} \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{x}}_L(k) \end{bmatrix} \quad (3.12)$$

Note that the Householder transformation allows normalization without the need of multiplication by a projection matrix, as it is required for the NCLMS in (2.12).

Figure 3.1 illustrates the coefficient update for the HCLMS and the NHCLMS algorithms, where as in Chapter 2, \mathcal{H}_0 and \mathcal{H}_1 are the hyperplanes defined by the null *a posteriori* condition and the constraints, respectively. Note that in this figure a rotation by θ is performed on the coordinate system, $\bar{\mathbf{w}} = [\bar{w}_1 \ \bar{w}_2]^T = \mathbf{Q}\mathbf{w}(k) = \mathbf{Q}[w_1 \ w_2]^T$. This angle is chosen such that the rotated axis \bar{w}_2 becomes parallel to the constraint hyperplane and the coordinate corresponding to \bar{w}_1 needs no further update. This is so because \bar{w}_1 becomes orthogonal to \mathcal{H}_1 . Table 3.1 shows an algorithmic description of the HCLMS algorithm.

3.2.3 Computational Complexity Issues

In this subsection we explain why and how the implementation via Householder transformation is better than the GSC and the constrained alternatives. Let us start with the procedure used to compute the product $\mathbf{Q}\mathbf{x}(k)$. In order to have an efficient Householder implementation, the transformation of the input-signal vector in every iteration is carried out through p reflections given by

$$\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k) = \mathbf{Q}_p \cdots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{x}(k) \quad (3.13)$$

where

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I}_{i-1 \times i-1} & \mathbf{0}^T \\ \mathbf{0} & \bar{\mathbf{Q}}_i \end{bmatrix} \quad (3.14)$$

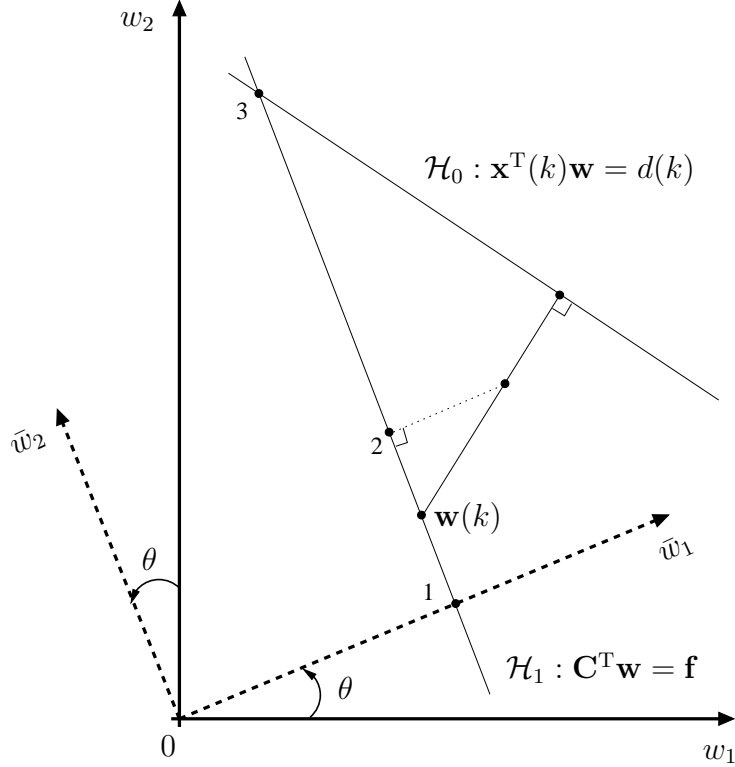


Figure 3.1: Coefficient-vector rotation.

1. $\bar{\mathbf{w}}(0) = \mathbf{Q}\mathbf{f} = \mathbf{Q}\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$.
2. $\mathbf{w}(k+1)$ for the HCLMS algorithm.
3. $\mathbf{w}(k+1)$ for the HNCLMS algorithm.

and matrix $\bar{\mathbf{Q}}_i = \mathbf{I} - 2\bar{\mathbf{v}}_i\bar{\mathbf{v}}_i^T$ is a $(MN - i + 1) \times (MN - i + 1)$ Householder transformation matrix [94].

If we define the vector $\mathbf{v}_i^T = [\mathbf{0}_{i-1}^T \bar{\mathbf{v}}_i^T]^T$, where the $p \times 1$ vector $\mathbf{0}_{i-1}$ introduces $i - 1$ zeros before $\bar{\mathbf{v}}_i$, we can construct the matrix $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_p]$, and the factored product in (3.13) could be implemented with the procedure described in Table 3.2. Furthermore, the procedure for the calculation of the Householder vectors and the resulting \mathbf{V} is described in Table 3.3, where \mathbf{A} is the matrix to be triangularized and in the particular case of interest, $\mathbf{A} = \mathbf{C}\mathbf{L}$ with \mathbf{L} being the square-root factor of the matrix $(\mathbf{C}^T\mathbf{C})^{-1}$ as proposed earlier in this section.

From Table 3.2 we see that the computation of $\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}_k$ using the product representation in (3.13) only involves $2MNp - p(p - 1)$ multiplications and $2MNp - p^2$ additions. Table 3.4 shows the computational complexity for the CLMS, NCLMS, HCLMS, NHCLMS

Table 3.1: The HCLMS Algorithm

Available at time instant k :
 $\mathbf{x}(k)$, \mathbf{C} , \mathbf{f} , \mathbf{Q} , and μ (step size)

Initialize:
 $\bar{\mathbf{w}}(0) = \mathbf{Q}\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$;

for $k = 0, 1, 2, \dots$

{

$\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k)$;

$\bar{\mathbf{x}}_L(k) = MN - p$ last elements of $\bar{\mathbf{x}}(k)$;

$\bar{\mathbf{w}}(k) = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k) \end{bmatrix}$;

$e(k) = d(k) - \bar{\mathbf{w}}^T(k)\bar{\mathbf{x}}(k)$;

$\bar{\mathbf{w}}_L(k+1) = \bar{\mathbf{w}}_L(k) + \mu e(k)\bar{\mathbf{x}}_L(k)$;

}

Table 3.2: Computation of $\mathbf{Q}\mathbf{x}(k)$

$\bar{\mathbf{x}}_k = \mathbf{x}(k)$;

for $i = 1 : p$

{

$\bar{\mathbf{x}}_k(i : MN) = \bar{\mathbf{x}}_k(i : MN)$
 $-2\mathbf{V}(i : MN, i) [\mathbf{V}^T(i : MN, i)\bar{\mathbf{x}}_k(i : MN)]$;

}

$\bar{\mathbf{x}}(k) = \bar{\mathbf{x}}_k$;

algorithms and the GSC implementation of the CLMS and NCLMS algorithms. The computational complexity for the GSC implementation is given for two choices of the blocking matrix \mathbf{B} . The first implementation, uses a \mathbf{B} matrix obtained by SVD leading to an inefficient implementation of the multiplication $\mathbf{B}\mathbf{x}(k)$. The second implementation applicable in certain problems, uses a \mathbf{B} matrix constructed through a sequence of sparse matrices as presented in [92], rendering an implementation of the multiplication $\mathbf{B}\mathbf{x}(k)$ of low computational complexity.

Table 3.3: Construction of Matrix \mathbf{V} Containing the Householder Vectors

Available at start:
 \mathbf{A} is an $MN \times p$ matrix to be triangularized
Initialize:
 $\mathbf{V} = \mathbf{0}_{MN \times p}$;

for $i = 1 : p$
{
 $\mathbf{x} = \mathbf{A}(i : MN, i)$;
 $\mathbf{e}_1 = [1 \ \mathbf{0}_{1 \times (MN-i)}]^T$
 $\mathbf{v} = \text{sign}(\mathbf{x}(1)) \|\mathbf{x}\| \mathbf{e}_1 + \mathbf{x}$;
 $\mathbf{v} = \mathbf{v} / \|\mathbf{v}\|$;
 $\mathbf{A}(i : MN, i : p) = \mathbf{A}(i : MN, i : p) - 2\mathbf{v} (\mathbf{v}^T \mathbf{A}(i : MN, i : p))$;
 $\mathbf{V}(i : MN, i) = \mathbf{v}$;
}

3.2.4 Householder-Transform Constrained Algorithms and the GSC

Figure 3.2 shows, step-by-step, the relation between a Householder-constrained (HC) algorithm and the GSC structure. If \mathbf{Q} is factored into an upper part and lower part (see Figure 3.2) it is easy to show that \mathbf{Q}_L , spans the null space of \mathbf{C} and may be regarded as a valid blocking matrix (see Appendix A3.1). Furthermore, $\mathbf{Q}_U^T \bar{\mathbf{w}}_U(0) = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{f} = \mathbf{F}$ (see Appendix A3.1), which is the upper part of the GSC structure. However, we stress that for most practical values of p the implementation of \mathbf{Q}_L and \mathbf{Q}_U separately renders much higher computational complexity because it does not take advantage of the efficiency of the Householder transformation. The transformed input-signal vector can be efficiently obtained via p Householder transformations which require only p inner products. We maintain that our approach can be regarded as a GSC structure and, therefore, any unconstrained adaptive algorithm can be used to update $\bar{\mathbf{w}}_L(k)$. As an example of this assertion, Table 3.5 shows the equations of the Householder-Transform Constrained Quasi-Newton (HCQN) algorithm obtained directly from [87] and Figure 3.2 as previously

Table 3.4: Computational Complexity

| ALGORITHM | ADD. | MULT. | DIV. |
|------------------------------------|------------------------------------|---------------------------------|------|
| CLMS | $(2p + 2)MN - (p + 1)$ | $(2p + 2)MN + 1$ | 0 |
| NCLMS | $(3p + 3)MN - (p + 2)$ | $(3p + 3)MN + 1$ | 1 |
| GSC-LMS | $(MN)^2 + (2 - p)MN - (1 + p)$ | $(MN)^2 + (3 - p)MN - (2p - 1)$ | 0 |
| GSC-NLMS | $(MN)^2 + (3 - p)MN - 2(1 + p)$ | $(MN)^2 + (4 - p)MN - (3p - 1)$ | 1 |
| GSC-LMS with \mathbf{B} of [92] | $(3 + p)MN - \frac{p(p+5)}{2} - 1$ | $(3 + 2p)MN - p(p + 3) + 1$ | 0 |
| GSC-NLMS with \mathbf{B} of [92] | $(4 + p)MN - \frac{p(p+7)}{2} - 2$ | $(4 + 2p)MN - p(p + 4) + 1$ | 1 |
| HCLMS | $(2p + 2)MN - (p^2 + p + 1)$ | $(2p + 2)MN - (p^2 - 1)$ | 0 |
| NHCLMS | $(2p + 3)MN - (p^2 + 2p + 2)$ | $(2p + 3)MN - (p^2 + p - 1)$ | 1 |

reported in [97]. Notice that the algorithm in Table 3.5 does not require the inversion and construction of the $p \times p$ matrix encountered in the conventional CQN algorithm presented in [87], resulting in a much simpler implementation of the algorithm.

3.3 Simulation Results

In this section the performances of the proposed algorithms is evaluated through simulations and compared to their GSC counterparts.

The same setup is used as in Section 2.7, where a uniform linear array with $M = 12$ antennas with element spacing equal to half wave-length was used in a system with $K = 5$ users, where the signal of one user is of interest and the other 4 are treated as interferers. The direction of arrival (DOA) and the signal-to-noise ratio (SNR) for the different signals is reproduced in Table 3.6. A second-order derivative constraint matrix [37] was used giving a total of three constraints. For the GSC implementation the nonunitary blocking matrix in [92] was used (see also Equation (2.46) in Chapter 2).

Figure 3.3 shows the learning curves of the different algorithms. The results were obtained by averaging 2000 realizations of the experiment. The step sizes used in the algorithms were $\mu = 5 \cdot 10^{-4}$ for the CLMS and the HCLMS algorithms, $\mu = 10^{-5}$ for the GSC-LMS algorithm, $\mu_n = 0.05$ for the NLMS algorithms, and $\alpha = 0.05$ for the QN algorithms.

As can be seen from the figure, the Householder implementations have a better perfor-

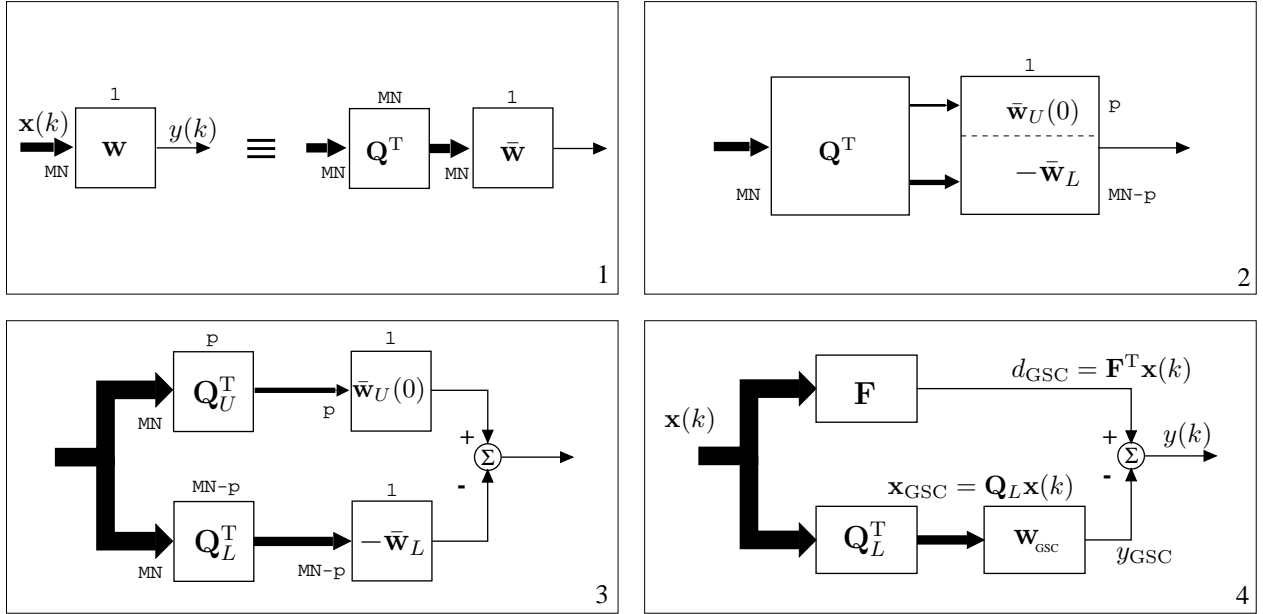


Figure 3.2: The HC adaptive filter under the GSC model.

1. Applying the transformation as in (3.1);
2. Splitting the transformed vector as in (3.5);
3. Partitioning Q in order to reach the GSC equivalent;
4. The HC algorithm viewed from a GSC perspective.

mance than the corresponding GSC implementations using the sparse blocking matrix.

Figures 3.4–3.6 show the beampatterns resulting from the different algorithms. The beampatterns obtained with the Householder algorithms are very close to the optimal solution. On the other hand, the GSC-based implementations failed to suppress completely all interferers at the same time, which suggests that the adaptation algorithms did not achieve a steady-state even after 7000 iterations. The output gains in the directions of the interferers are shown in Table 3.7.

3.4 Conclusions

In this chapter we presented an efficient implementation of linearly-constrained minimum-variance adaptive filters based on the Householder transformation of the input signal.

Table 3.5: The HCQN Algorithm

Available at time instant k :
 $\mathbf{x}(k)$, \mathbf{C} , \mathbf{f} , and \mathbf{Q}

Initialize:
 α , $\bar{\mathbf{R}}_L^{-1}(0)$, and $\bar{\mathbf{w}}(0) = \mathbf{QC}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$

for $k = 1, 2, \dots$

{

$\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k)$;
 $\bar{\mathbf{x}}_U(k) = p$ first elements of $\bar{\mathbf{x}}(k)$;
 $\bar{\mathbf{x}}_L(k) = MN - p$ last elements of $\bar{\mathbf{x}}(k)$;
 $\bar{\mathbf{w}}(k-1) = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ -\bar{\mathbf{w}}_L(k-1) \end{bmatrix}$;
 $\bar{e}(k) = \bar{\mathbf{w}}_U^T(0)\bar{\mathbf{x}}_U(k) - \bar{\mathbf{w}}_L^T(k-1)\bar{\mathbf{x}}_L(k)$;
% $\bar{e}(k)$ is equivalent to the *a priori* output
% or $y(k) = \bar{\mathbf{w}}^T(k-1)\bar{\mathbf{x}}(k)$
 $\mathbf{t}(k) = \bar{\mathbf{R}}_L^{-1}(k-1)\bar{\mathbf{x}}_L(k)$;
 $\tau = \bar{\mathbf{x}}_L^T(k)\mathbf{t}(k)$;
 $\mu(k) = \frac{1}{2\tau(k)}$;
 $\bar{\mathbf{R}}_L^{-1}(k) = \bar{\mathbf{R}}_L^{-1}(k-1) + \frac{\mu(k)-1}{\tau(k)}\mathbf{t}(k)\mathbf{t}^T(k)$;
 $\bar{\mathbf{w}}_L(k) = \bar{\mathbf{w}}_L(k-1) + \alpha\frac{\bar{e}(k)}{\tau(k)}\mathbf{t}(k)$;

}

With this type of transformation, we derived several adaptation algorithms for LCMV applications, such as the Householder-transform constrained least mean square algorithm and its normalized version, and maintained that extension to other adaptation algorithms should be trivial.

Via Householder transformation we were able to reduce the dimension of the subspace in which the adaptive-filter coefficients are updated, therefore obtaining a transformed input signal which is persistently exciting. Viewed under the perspective of the generalized sidelobe canceling model, we showed that the transformation matrix can be factored into a matrix satisfying the constraints and a blocking matrix.

In terms of computational complexity our method is comparable to the most efficient implementations of the blocking matrix found in the literature, with the advantage that the Householder transformation, and consequently the blocking matrix implicitly used in

Table 3.6: Signal Parameters

| SIGNAL | DOA θ_i | SNR |
|--------------|----------------|-------|
| desired | 0° | 15 dB |
| interferer 1 | 22° | 20 dB |
| interferer 2 | -15° | 25 dB |
| interferer 3 | -20° | 25 dB |
| interferer 4 | -50° | 20 dB |

Table 3.7: Output Gains in the Directions of the Interferers.

| ALGORITHM | $\theta = 22^\circ$ | $\theta = -15^\circ$ | $\theta = -20^\circ$ | $\theta = -50^\circ$ |
|-----------|---------------------|----------------------|----------------------|----------------------|
| GSC-LMS | -26.20 dB | -13.29 dB | -11.76 dB | -22.99 dB |
| GSC-NLMS | -31.74 dB | -21.39 dB | -18.73 dB | -22.67 dB |
| GSC-QN | -26.97 dB | -27.29 dB | -22.79 dB | -24.88 dB |
| HCLMS | -35.40 dB | -23.38 dB | -17.94 dB | -32.85 dB |
| NHCLMS | -31.50 dB | -33.85 dB | -24.82 dB | -27.83 dB |
| HCQN | -32.92 dB | -30.76 dB | -28.66 dB | -26.75 dB |

the transformation, are unitary. For this reason, not only the steady-state mean squared output error is the same as that of the conventional nontransformed LCMV filter, but the equivalence is also verified during the transient. Having a unitary transformation also imparts robustness to the method, especially when applied to nonconventional Wiener-filter structures (e.g., multistage representation). Some of these properties were illustrated in one example of beamforming.

Appendix A3.1

This Appendix contains the details related to the discussion on the Householder Transform algorithms under a GSC perspective. Let \mathbf{Q} be partitioned as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_U \\ \dots \\ \mathbf{Q}_L \end{bmatrix}$$

where \mathbf{Q}_U has dimension $p \times MN$ and \mathbf{Q}_L has dimension $(MN - p) \times MN$. It will be shown below that by using this partition of \mathbf{Q} , a GSC structure can be derived. It turns out that \mathbf{Q}_U can be related to the upper part in the GSC, which consist of the filter \mathbf{F} , and,

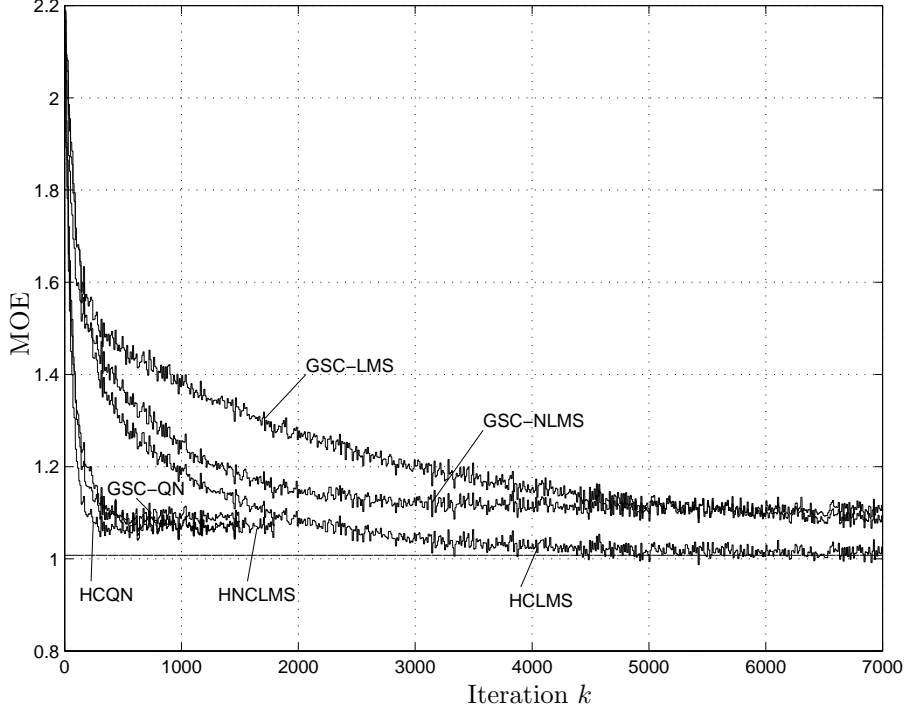


Figure 3.3: Learning curves of the algorithms.

that \mathbf{Q}_L can be regarded as a valid blocking matrix. First, we show that $\mathbf{Q}_U^T \bar{\mathbf{w}}_L(0) = \mathbf{F}$. Pre-multiply Equation (3.2) with \mathbf{Q}^T

$$\begin{aligned} \mathbf{Q}^T \bar{\mathbf{C}} (\bar{\mathbf{C}}^T \bar{\mathbf{C}})^{-1} \bar{\mathbf{C}}^T &= \mathbf{C} (\mathbf{C}^T \mathbf{C})^{-1} \bar{\mathbf{C}}^T \\ &= \mathbf{Q}^T \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = [\mathbf{Q}_U^T \ \mathbf{0}] \end{aligned} \quad (3.15)$$

where the result from Equation (3.2) was used together with the equality $\bar{\mathbf{C}}^T \bar{\mathbf{C}} = \mathbf{C}^T \mathbf{Q}^T \mathbf{Q} \mathbf{C} = \mathbf{C}^T \mathbf{C}$. Post-multiplying (3.15) with $\bar{\mathbf{w}}(k)$ gives

$$\begin{aligned} \mathbf{C} (\mathbf{C}^T \mathbf{C})^{-1} \bar{\mathbf{C}}^T \bar{\mathbf{w}}(k) &= \mathbf{C} (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{w}(k) = \mathbf{F} \\ &= [\mathbf{Q}_U^T \ \mathbf{0}] \bar{\mathbf{w}}(k) = [\mathbf{Q}_U^T \ \mathbf{0}] \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \dots \\ \bar{\mathbf{w}}_L(k) \end{bmatrix} \\ &= \mathbf{Q}_U^T \bar{\mathbf{w}}_U(0) \end{aligned} \quad (3.16)$$

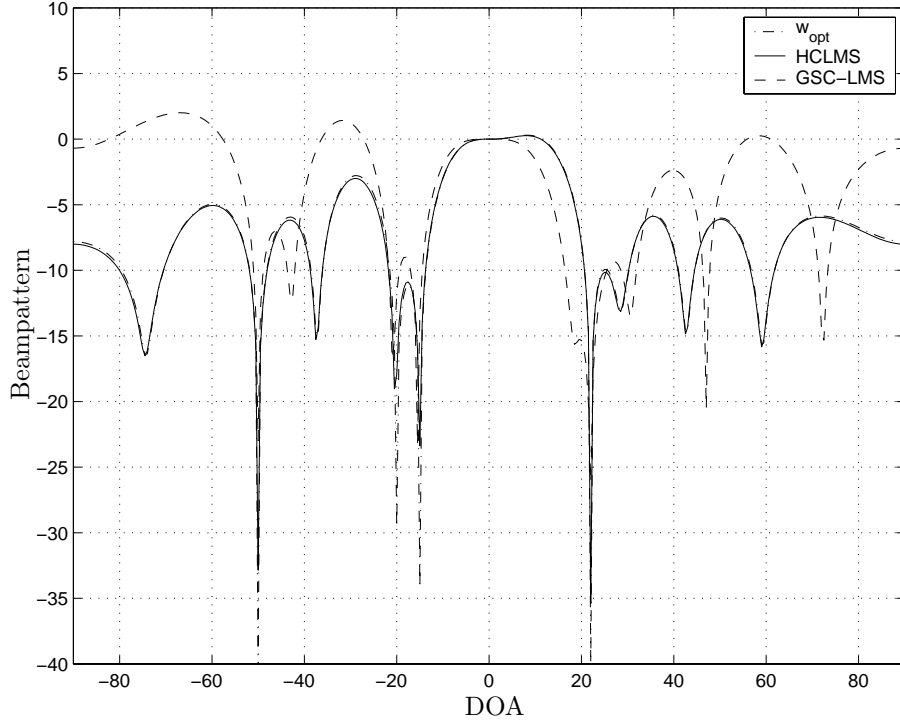


Figure 3.4: Beampattern for the HCLMS and GSC-LMS algorithms.

which is the the upper part in the GSC structure. In order to show that \mathbf{Q}_L constitutes a valid blocking matrix, note that

$$\begin{aligned}
 \mathbf{Q}\mathbf{C} &= \begin{bmatrix} \mathbf{Q}_U\mathbf{C} \\ \dots \\ \mathbf{Q}_L\mathbf{C} \end{bmatrix} = \bar{\mathbf{C}}(\bar{\mathbf{C}}^T\bar{\mathbf{C}})^{-1}\bar{\mathbf{C}}^T\bar{\mathbf{C}} \\
 &= \begin{bmatrix} \mathbf{I}_{p \times p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{Q}\mathbf{C} = \begin{bmatrix} \mathbf{Q}_U\mathbf{C} \\ \dots \\ \mathbf{0} \end{bmatrix} \tag{3.17}
 \end{aligned}$$

and, therefore, $\mathbf{Q}_L\mathbf{C} = \mathbf{0}$, and together with the fact that \mathbf{Q}_L has full rank we can conclude that it fulfills the requirement for a valid blocking matrix.

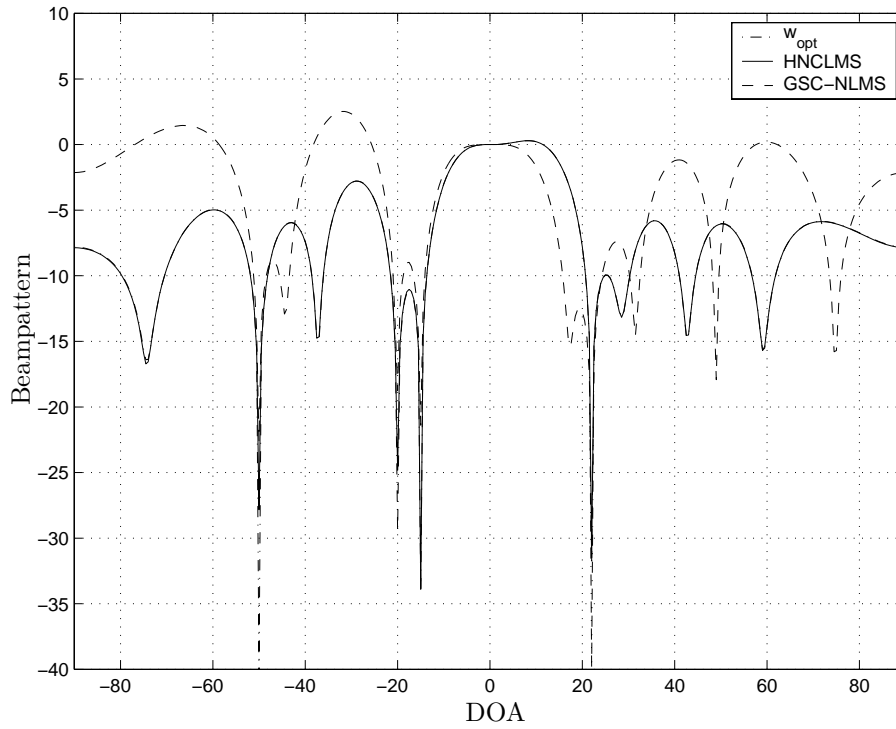


Figure 3.5: Beampattern for the HNCLMS and the GSC-NLMS algorithms.

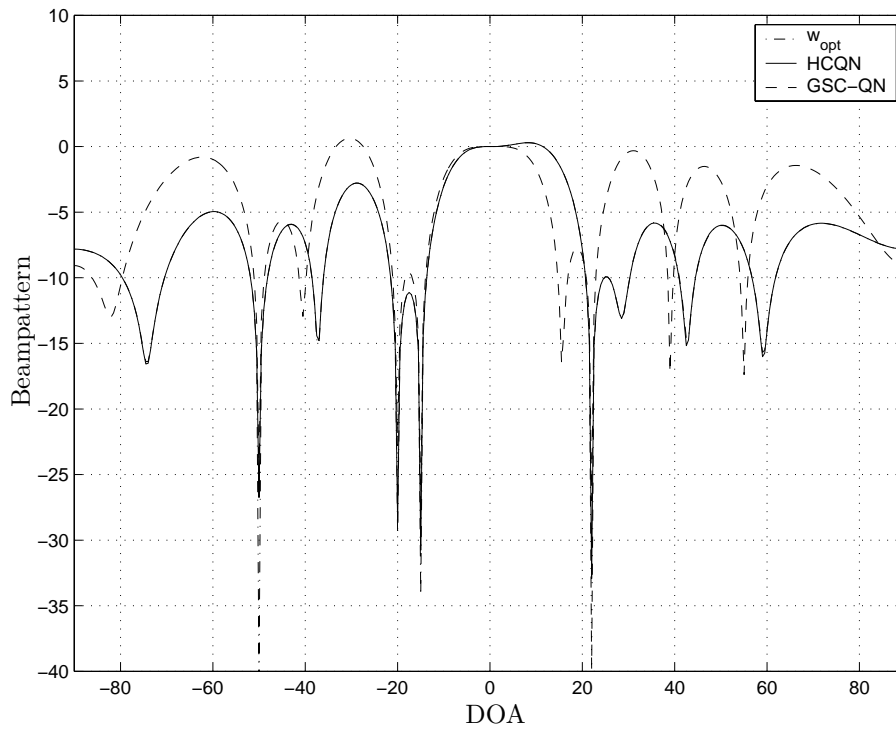


Figure 3.6: Beampattern for the HCQN and the GSC-QN algorithms.

Chapter 4

Set-Membership Binormalized Data Reusing LMS Algorithms

This chapter presents and analyzes novel data selective normalized adaptive filtering algorithms with two data reuses. The algorithms, the set-membership binormalized LMS (SM-BNDRLMS) algorithms, are derived using the concept of set-membership filtering (SMF). These algorithms can be regarded as generalizations of the recently proposed set-membership NLMS (SM-NLMS) algorithm. They include two constraint sets in order to construct a space of feasible solutions for the coefficient updates. The algorithms include data-dependent step sizes that provide fast convergence and low excess mean-squared error (MSE). Convergence analyses in the mean squared sense are presented and closed-form expressions are given for both white and colored input signals. A study of the transient of the SM-NLMS algorithm is performed, which suggests a slowdown in convergence speed for colored input signals. Simulation results show the good performance of the new algorithms in terms of convergence speed, final misadjustment, and reduced computational complexity.

4.1 Introduction

The least mean square (LMS) algorithm has gained popularity due to its robustness and low computational complexity. The main drawback of the LMS algorithm is that the con-

vergence speed depends strongly on the eigenvalue spread of the input-signal correlation matrix [1]. To overcome this problem, a more complex recursive least squares (RLS) type of algorithm can be used. However, the faster convergence of the RLS algorithm does not imply a better tracking capability in a time-varying environment [1]. An alternative to speed up the convergence at the expense of low additional complexity is to use the binormalized data-reusing LMS (BNDRLMS) algorithm [51, 52]. The BNDRLMS algorithm, which uses consecutive data pairs in each update, has shown fast convergence for correlated input signals. However, the fast convergence comes at the expense of higher misadjustment, because the algorithm utilizes the data even if it does not imply innovation. In order to combat the conflicting requirements of fast convergence and low misadjustment, the objective function of the adaptive algorithm needs to be changed. Set-membership filtering (SMF) [68, 69, 70, 98, 99] specifies a bound on the magnitude of the estimation error. The SMF uses the framework of set-membership identification (SMI) [100, 101, 102, 103] to include a general filtering problem. Consequently, many of the existing optimal bounding ellipsoid (OBE) algorithms [72, 100, 104, 105, 106] can be applied to the SMF framework.

Most, if not all, of the SMF algorithms feature reduced computational complexity primarily due to (sparse) *data-selective* updates. Implementation of those algorithms essentially involves two steps: (1) information evaluation (innovation check); and (2) update of parameter estimate. If the update does not occur frequently and the information evaluation does not involve much computational complexity, the overall complexity is usually much less than that of their RLS (as well as LMS) counterparts. It was shown in [68] that the class of adaptive solutions, called *set-membership adaptive recursive techniques* (SMART) include a particularly attractive OBE algorithm, referred to as the Quasi-OBE algorithm or the bounding ellipsoidal adaptive constrained least-squares (BEACON) algorithm [70, 72], with a complexity of $O(N)$ for the innovation check. Also in [68] an algorithm with recursions similar to those of the NLMS algorithm with an adaptive step size was derived. The algorithm named set-membership NLMS (SM-NLMS) algorithm, further studied in [69], was shown to achieve both fast convergence and low misadjustment. Applications of set-membership filtering include adaptive equalization where it allows the

sharing of hardware resources in multichannel communications systems [70], adaptive multiuser detection in CDMA systems [71, 107], and in filtering with deterministic constraints on the output-error sequence [108].

The SM-NLMS algorithm only uses the current input signal in its update. We show that the convergence speed of the SM-NLMS algorithm depends on the eigenvalue spread of the input signal. In order to overcome this problem we propose two versions of an algorithm that uses data pairs from two successive time instants in order to construct a set of feasible solutions for the update. The new algorithms are also data-selective algorithms leading to a low average computational complexity per update. In addition, for correlated input signals, they retain the fast convergence of the BNDRLMS algorithms related to the smart reuse of input-desired data pairs. The low misadjustment is obtained due to the data-selective updating utilized by the new algorithms. The idea of data reuse was also exploited in the context of OBE algorithms in [106].

The organization of the chapter is as follows. Section 4.2 reviews the concept of set-membership filtering, and the SM-NLMS algorithm of [69]. In Section 4.2 we also study the convergence speed of the SM-NLMS algorithm. The new algorithms are derived in Section 4.3. Section 4.4 contains analysis of the algorithms in the mean-squared sense, followed by simulations in Section 4.5. Section 4.6 contains the concluding remarks.

4.2 Set-Membership Filtering

This section reviews the basic concepts of set-membership filtering (SMF). For a more detailed introduction to the concept of SMF, the reader is referred to [70]. Set-membership filtering (SMF) is a framework applicable to filtering problems. A specification on the filter parameters is achieved by constraining the output estimation error to be smaller than a deterministic threshold. As a result of the bounded error constraint there will exist a set of filters rather than a single estimate. The SMF paradigm is inspired by set-membership identification (SMI) applicable in system-identification when a *bounded-noise* assumption can be made. We will not discuss SMI in this thesis and, for an extensive treatment, we

refer to [101, 102, 105]. SMF, with its *bounded-error* specification, finds several applications in signal processing for communications where a bounded noise assumption cannot be made and where the assumption of the existence of “true parameters” is unnatural. Examples of such applications are equalization [70, 107, 109], adaptive multiuser detection [71] and beamforming [110].

In SMF, the filter \mathbf{w} is designed to achieve a specified bound γ on the magnitude of the output error. This bound is a design parameter and can vary depending on the application. For example, it was shown in [70] that perfect equalization is obtained if $\gamma = d_{min}/2$ where d_{min} is the minimum distance between the signal constellations.

Assuming a sequence of input vectors $\{\mathbf{x}(k)\}_{k=1}^{\infty}$ and a desired signal sequence $\{d(k)\}_{k=1}^{\infty}$, we can write the sequence of estimation errors $\{e(k)\}_{k=1}^{\infty}$ as,

$$e(k) = d(k) - \mathbf{w}^T \mathbf{x}(k) \quad (4.1)$$

where $\mathbf{x}(k)$ and $\mathbf{w} \in \mathbb{R}^N$ with $d(k)$ and $e(k) \in \mathbb{R}$. For a properly chosen bound γ on the estimation error, there are infinitely many valid estimates of \mathbf{w} .

Let \mathcal{S} denote the set of all possible input-desired data pairs (\mathbf{x}, d) of interest. Let Θ denote the set of all possible vectors \mathbf{w} that result in an output error bounded by γ whenever $(\mathbf{x}, d) \in \mathcal{S}$. The set Θ referred to as the *feasibility set* is given by

$$\Theta = \bigcap_{(\mathbf{x}, d) \in \mathcal{S}} \{\mathbf{w} \in \mathbb{R}^N : |d - \mathbf{w}^T \mathbf{x}| \leq \gamma\} \quad (4.2)$$

Assume that the adaptive filter is trained with k input-desired data pairs $\{\mathbf{x}(i), d(i)\}_{i=1}^k$. Let $\mathcal{H}(k)$ denote the set containing all vectors \mathbf{w} for which the associated output error at time instant k is upper bounded in magnitude by γ . In other words,

$$\mathcal{H}(k) = \{\mathbf{w} \in \mathbb{R}^N : |d(k) - \mathbf{w}^T \mathbf{x}(k)| \leq \gamma\} \quad (4.3)$$

The set $\mathcal{H}(k)$ is referred to as the *constraint set* and its boundaries are hyperplanes. Finally define the *exact membership set* ψ_k to be the intersection of the constraint sets over the time instants $i = 1, \dots, k$, i.e.,

$$\psi(k) = \bigcap_{i=1}^k \mathcal{H}(i) \quad (4.4)$$

It can be seen that the *feasibility set* Θ is a subset of the *exact membership set* ψ_k at any given time instant. The *feasibility set* is also the *limiting set* of the *exact membership set*, i.e., the two sets will be equal if the training signal traverses all signal pairs belonging to \mathcal{S} .

The idea of SMART is to adaptively find an estimate that belongs to the feasibility set or to one of its members. Since $\psi(k)$ in (4.4) is an N dimensional polytope, it is not easily computed. One approach is to apply one of the many optimal bounding ellipsoid (OBE) algorithms, which tries to approximate the exact membership set $\psi(k)$ by tightly outer bound it with ellipsoids. Another adaptive approach is to compute a point estimate through projections using, for example, the information provided by the constraint set $\mathcal{H}(k)$ like in the set-membership NLMS (SM-NLMS) algorithm considered in the following subsection. It was also shown in [69] that the SM-NLMS algorithm can be associated with an optimal bounding spheroid (OBS).

4.2.1 Set-Membership Normalized LMS Algorithm

The set-membership NLMS (SM-NLMS) algorithm derived in [69] is similar to the conventional NLMS algorithm in form. However, the philosophy behind the SM-NLMS algorithm derivation differs from that of the NLMS algorithm. The basic idea behind the algorithm is that if the previous estimate $\mathbf{w}(k)$ lies outside the constraint set $\mathcal{H}(k)$, i.e., $|d(k) - \mathbf{w}^T(k)\mathbf{x}(k)| > \gamma$, the new estimate $\mathbf{w}(k+1)$ will lie on the closest boundary of $\mathcal{H}(k)$ at a minimum distance, i.e., the SM-NLMS minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint that $\mathbf{w}(k+1) \in \mathcal{H}(k)$. This is obtained by an orthogonal projection of the previous estimate onto the closest boundary of $\mathcal{H}(k)$. A graphical visualization of the updating procedure of the SM-NLMS can be found in Figure 4.1. Straightforward calculation leads to the following recursions for $\mathbf{w}(k)$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k) \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \quad (4.5)$$

with

$$\alpha(k) = \begin{cases} 1 - \frac{\gamma}{|e(k)|}, & \text{if } |e(k)| > \gamma \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

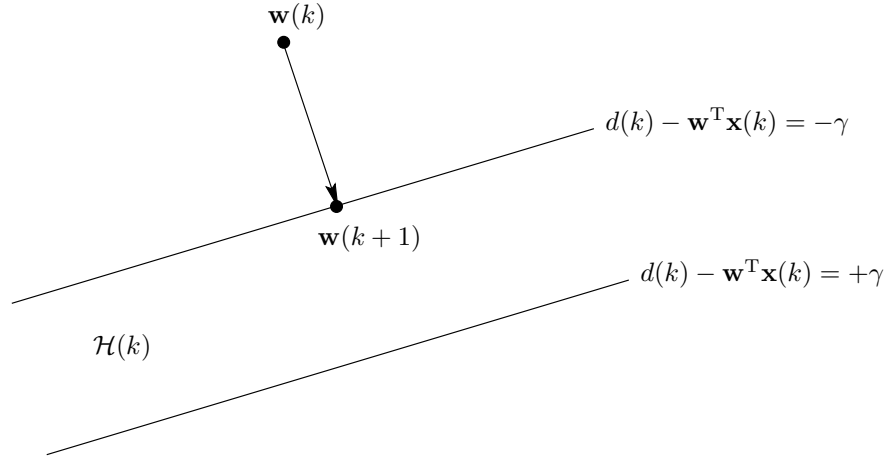


Figure 4.1: The SM-NLMS algorithm.

where $\alpha(k)$ denotes the time-dependent step size. The update equations (4.5)–(4.6) resemble those of the conventional NLMS algorithm except for the time-varying step size $\alpha(k)$. Note that, since the conventional NLMS algorithm minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint that $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$, it is a particular case of the above algorithm by choosing the bound $\gamma = 0$. Furthermore, using a step size $\alpha(k) = 1$ in the SM-NLMS whenever $\mathbf{w}(k) \notin \mathcal{H}(k)$, would result in a valid update because the hyperplane with zero *a posteriori* error lies in $\mathcal{H}(k)$. However, the resulting algorithm does not minimize the Euclidean distance of the coefficient-vector update.

4.2.2 SM-NLMS Algorithm – Convergence Issues

The SM-NLMS algorithm reviewed in the previous section has been proven to have the following important features [69]:

- the magnitude of the parameter error $\mathbf{w}(k+1) - \mathbf{w}_*$, where \mathbf{w}_* is any point in the feasibility set, is monotonically nonincreasing;
- the magnitude of the difference of the parameter estimates between consecutive iterations $\mathbf{w}(k+1) - \mathbf{w}(k)$ converges to zero, and;
- the step size converges to zero, and the magnitude of the prediction error is asymptotically smaller than γ .

The mentioned convergence features are indeed important but it would be of interest to get an idea about how the convergence speed is affected by the statistics of the input signal. Simulations indicate that the convergence speed of the SM-NLMS algorithm depends on the eigenvalue spread of the input-signal correlation matrix. Due to the similarity in form to the conventional NLMS algorithm it does not come as a surprise. However, no such study has yet been performed, and it is the objective of this section to make a quantitative study of the convergence speed of the SM-NLMS algorithm for the case of a large number of filter coefficients.

A number of assumptions are made to simplify our study: (1) independence assumption [1]; (2) a large number of coefficients in the adaptive filter, and; (3) the SM-NLMS algorithm performs an update at time instant k with the probability $P_e(k) = Pr\{|e(k)| > \gamma\}$.

The coefficient error at time instant $k + 1$ defined as $\Delta\mathbf{w}(k + 1) = \mathbf{w}(k) - \mathbf{w}_{opt}$, is given by

$$\Delta\mathbf{w}(k + 1) = \left[\mathbf{I} - \alpha(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \Delta\mathbf{w}(k) + \alpha(k) \frac{n(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \quad (4.7)$$

Invoking the independence assumption and using some results of Section 4.4 we get

$$\mathbb{E}[\Delta\mathbf{w}(k + 1)] = \mathbb{E} \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \mathbb{E}[\Delta\mathbf{w}(k)] \quad (4.8)$$

If N is assumed large and $\mathbb{E}[x(k)] = 0$, we have $\|\mathbf{x}(k)\|^2 \approx \text{tr}\{\mathbf{R}\}$, which leads to

$$\mathbb{E}[\Delta\mathbf{w}(k + 1)] = \left[\mathbf{I} - \mathbb{E}\{P_e(k)\} \frac{\mathbf{R}}{\text{tr}\{\mathbf{R}\}} \right] \mathbb{E}[\Delta\mathbf{w}(k)] \quad (4.9)$$

Using the *spectral decomposition* of \mathbf{R} as $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ [1], where \mathbf{Q} is a unitary matrix whose columns are equal to the eigenvectors of \mathbf{R} and $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues, i.e., $\mathbf{\Lambda} = \text{diag}[\lambda_1 \dots \lambda_N]$. We can now define the rotated vector

$$\Delta \bar{\mathbf{w}}(k+1) = \mathbf{Q}^T \Delta \mathbf{w}(k+1),$$

$$\begin{aligned} \mathbb{E}[\Delta \bar{\mathbf{w}}(k+1)] &= \left[\mathbf{Q}^T - \mathbb{E}\{P_e(k)\} \frac{\mathbf{Q}^T \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T}{\text{tr}\{\mathbf{R}\}} \right] \mathbb{E}[\Delta \mathbf{w}(k)] \\ &= \left[\mathbf{I} - \mathbb{E}\{P_e(k)\} \frac{\mathbf{\Lambda}}{\text{tr}\{\mathbf{R}\}} \right] \mathbb{E}[\Delta \bar{\mathbf{w}}(k)] \\ &= \prod_{i=0}^k \left[\mathbf{I} - \mathbb{E}\{P_e(i)\} \frac{\mathbf{\Lambda}}{\text{tr}\{\mathbf{R}\}} \right] \mathbb{E}[\Delta \bar{\mathbf{w}}(0)] \\ &= \prod_{i=0}^k \begin{bmatrix} 1 - \frac{\mathbb{E}\{P_e(i)\}\lambda_1}{\sum_{j=1}^N \lambda_j} & 0 & \cdots & 0 \\ 0 & 1 - \frac{\mathbb{E}\{P_e(i)\}\lambda_2}{\sum_{j=1}^N \lambda_j} & 0 & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 - \frac{\mathbb{E}\{P_e(i)\}\lambda_N}{\sum_{j=1}^N \lambda_j} \end{bmatrix} \mathbb{E}[\Delta \bar{\mathbf{w}}(0)] \end{aligned} \quad (4.10)$$

For $\mathbb{E}\{P_e(k)\} \neq 0$ the coefficient-error will decrease, and the slowest mode will determine the speed. During the initial transient we have $\mathbb{E}\{P_e(k)\} \neq 0$ and we can, therefore, conclude that during the initial transient, the convergence speed of the SM-NLMS algorithm will depend on the eigenvalue spread of the input signal.

To improve the convergence speed at a small additional computational complexity per update, we will in next section study set-membership adaptation algorithms reusing two input-signal data-pairs.

4.3 Set-Membership Binormalized Data-Reusing LMS Algorithms

The SM-NLMS algorithm in the previous subsection only considered the constraint set $\mathcal{H}(k)$ in its update. The SM-NLMS algorithm has a low computational complexity per update but from previous section we could see that its convergence speed follows the trend of the normalized LMS algorithm which depends on the eigenvalue spread of the input-signal correlation matrix. The exact membership set $\psi(k)$ defined in (4.4) suggests the use of more than one constraint set. In this subsection, two algorithms are derived requiring that the solution belongs to the constraint sets at time instants k and $k-1$, i.e.,

$\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$. The recursions of the algorithms are similar to those of the conventional BNDRLMS algorithm [51]. The set-membership binormalized data-reusing LMS (SM-BNDRLMS) algorithms can be seen as extensions of the SM-NLMS algorithm that use two consecutive constraint sets for each update. The first algorithm presented in Section 4.3.1 is a two-step approach minimizing the Euclidean distance between the old filter coefficients and the new update subject to the constraints that the new update lies in both constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$. The second algorithm presented in Section 4.3.2 reduces the computational complexity per update as compared to the first algorithm by choosing a different update strategy.

4.3.1 Algorithm I

The first set-membership binormalized data-reusing LMS algorithm (SM-BNDRLMS-I) performs an initial normalized step according to the SM-NLMS algorithm. If the solution to the first step belongs to both constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ no further update is required. If the initial step moves the solution out of $\mathcal{H}(k-1)$, a second step is taken such that the solution is at the intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ at a minimum distance from $\mathbf{w}(k)$. Figure 2 depicts the update procedure. The SM-BNDRLMS-I algorithm minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to $\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$.

The solution can be obtained by first performing an orthogonal projection of $\mathbf{w}(k)$ onto the nearest boundary of $\mathcal{H}(k)$ just like in the SM-NLMS algorithm

$$\mathbf{w}'(k) = \mathbf{w}(k) + \alpha(k) \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \quad (4.11)$$

where $\alpha(k)$ is defined in (4.6) and $e(k)$ is defined in (4.1). If $\mathbf{w}'(k) \in \mathcal{H}(k-1)$, i.e., $|d(k-1) - \mathbf{w}'^T(k)\mathbf{x}(k-1)| \leq \gamma$, then $\mathbf{w}(k+1) = \mathbf{w}'(k)$. Otherwise a second step is taken such that the solution lies at the intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ at a minimum distance from the previous coefficient vector. The second step in the algorithm will be in the direction of $\mathbf{x}^\perp(k)$, which is orthogonal to the first step, i.e.,

$$\mathbf{w}(k+1) = \mathbf{w}'(k) + \beta(k) \frac{\epsilon(k-1)\mathbf{x}^\perp(k)}{\|\mathbf{x}^\perp(k)\|^2} \quad (4.12)$$

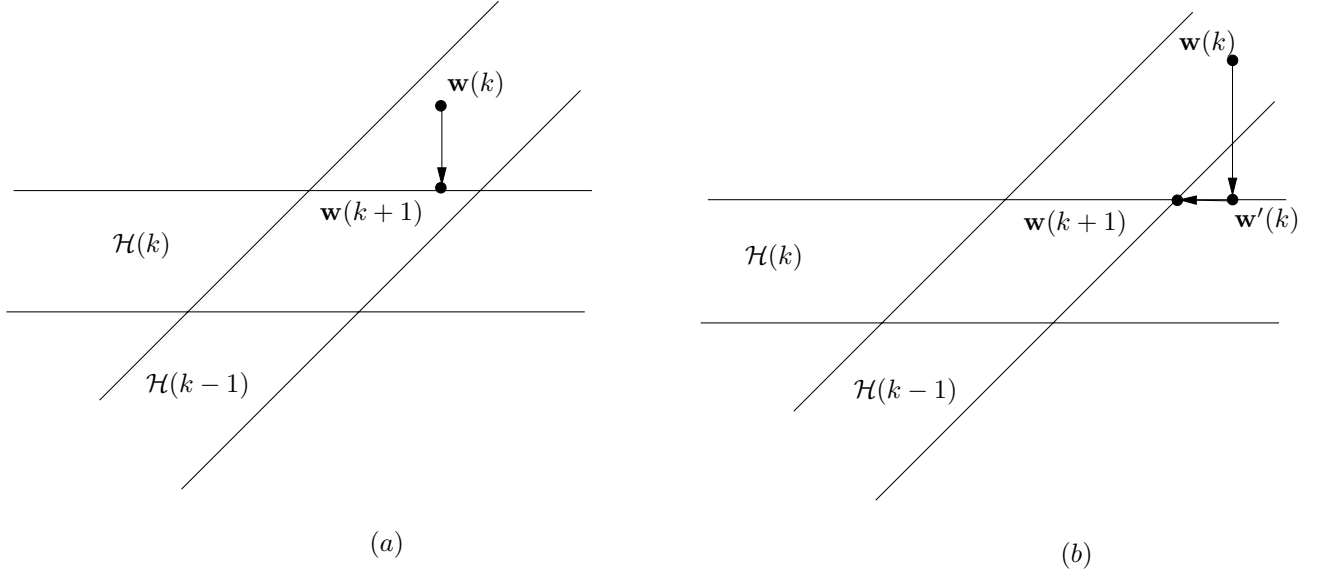


Figure 4.2: The SM-BNDRLMS-I algorithm: (a) The orthogonal projection onto the nearest boundary of $\mathcal{H}(k)$ lies within $\mathcal{H}(k-1)$, i.e., $\mathbf{w}'(k) \in \mathcal{H}(k-1)$, no further update. (b) The orthogonal projection onto the nearest boundary of $\mathcal{H}(k)$, $\mathbf{w}'(k)$, lies outside $\mathcal{H}(k-1)$, final solution at the nearest intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$.

where

$$\begin{aligned}
 \mathbf{x}^\perp(k) &= \left(\mathbf{I} - \frac{\mathbf{x}(k)\mathbf{x}^\top(k)}{\|\mathbf{x}(k)\|^2} \right) \mathbf{x}(k-1) \\
 \epsilon(k-1) &= d(k-1) - \mathbf{w}^\top(k)\mathbf{x}(k-1) \\
 \beta(k) &= 1 - \frac{\gamma}{|\epsilon(k-1)|}
 \end{aligned} \tag{4.13}$$

In summary, the recursive algorithm for $\mathbf{w}(k)$ is given by

$$\begin{aligned}
 \mathbf{w}'(k) &= \mathbf{w}(k) + \alpha(k) \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \\
 \mathbf{w}(k+1) &= \mathbf{w}'(k) + \lambda_1 \mathbf{x}(k) + \lambda_2 \mathbf{x}(k-1)
 \end{aligned} \tag{4.14}$$

where

$$\begin{aligned}
e(k) &= d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \\
\epsilon(k-1) &= d(k-1) - \mathbf{w}^T(k)\mathbf{x}(k-1) \\
\lambda_1 &= -\frac{\beta(k)\epsilon(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \\
\lambda_2 &= \frac{\beta(k)\epsilon(k-1)\|\mathbf{x}(k)\|^2}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \\
\alpha(k) &= \begin{cases} 1 - \frac{\gamma}{|e(k)|}, & \text{if } |e(k)| > \gamma \\ 0, & \text{otherwise} \end{cases} \\
\beta(k) &= \begin{cases} 1 - \frac{\gamma}{|\epsilon(k-1)|}, & \text{if } |e(k)| > \gamma \text{ and } |\epsilon(k-1)| > \gamma \\ 0 & \text{otherwise} \end{cases} \tag{4.15}
\end{aligned}$$

Remark. If the constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ are parallel, the denominator term of the λ_i s in (4.15) will be zero. In this particular case the second step of Equation (4.14) is not performed to avoid division with zero.

It is easy to verify that if the bound of the estimation error is chosen to be zero, i.e., $\gamma = 0$, the update equations will be those of the conventional BNDRLMS algorithm with unity a step size [51]. Table 4.1 shows the recursions of the SM-BNDRLMS-I algorithm.

Lemma 2. The magnitude of the parameter error $\|\mathbf{w}_* - \mathbf{w}(k+1)\|$, where \mathbf{w}_* is any point in the feasibility set, and $\mathbf{w}(k+1)$ is given by (4.14), is a monotonically nonincreasing sequence.

Proof. For the case where the SM-BNDRLMS-I only uses the first step, its recursions become equal to those of the SM-NLMS algorithm which has been shown in [69] to have $\|\mathbf{w}_* - \mathbf{w}(k+1)\| \leq \|\mathbf{w}_* - \mathbf{w}(k)\|$. For the case when both steps are taken (see Figure 4.2b), introduce the hyperplanes $\mathbf{w}_*^T\mathbf{x}(k) = d_*$ and $\mathbf{w}_*^T\mathbf{x}(k-1) = d_*$. Let \mathbf{w}_0 and \mathbf{w}_1 denote, respectively, the intersections of the extensions of $\mathbf{w}(k)$ and $\mathbf{w}(k+1)$ in the directions of $\mathbf{x}(k)$ with the hyperplane defined by $[\mathbf{w}_* - \mathbf{w}]^T\mathbf{x}(k)$, see Figure 4.3 for a graphical illustration. Using $[\mathbf{w}_* - \mathbf{w}_0] \perp [\mathbf{w}_0 - \mathbf{w}(k)]$, $[\mathbf{w}_* - \mathbf{w}_1] \perp [\mathbf{w}_1 - \mathbf{w}(k+1)]$, and $[\mathbf{w}_0 - \mathbf{w}(k)] \parallel [\mathbf{w}_1 - \mathbf{w}(k+1)]$

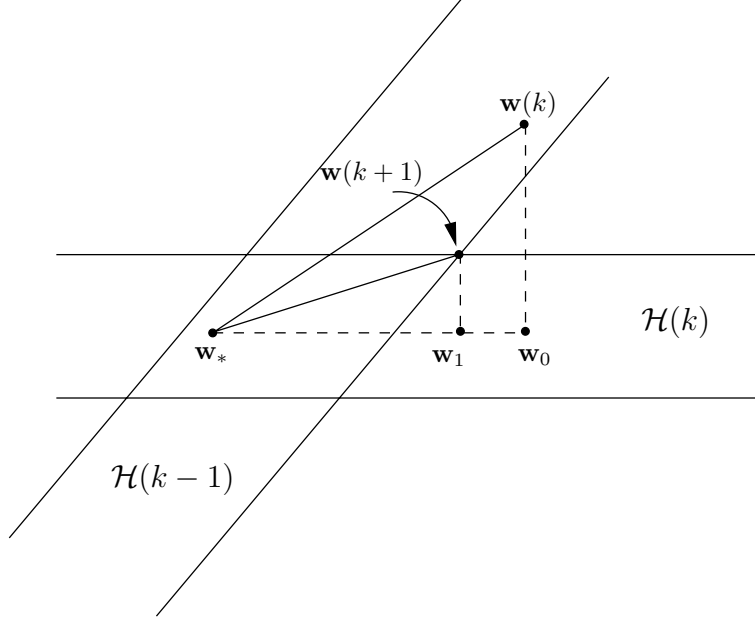


Figure 4.3: Distance evolution.

we get

$$\begin{aligned}\|\mathbf{w}_* - \mathbf{w}(k)\|^2 &= \|\mathbf{w}_0 - \mathbf{w}(k)\|^2 + \|\mathbf{w}_* - \mathbf{w}_0\|^2 \\ &\geq \|\mathbf{w}_1 - \mathbf{w}'(k)\|^2 + \|\mathbf{w}_* - \mathbf{w}_0\|^2\end{aligned}$$

since $\mathbf{w}(k) \notin \mathcal{H}(k-1)$, and

$$\begin{aligned}\|\mathbf{w}_* - \mathbf{w}(k+1)\|^2 &= \|\mathbf{w}_1 - \mathbf{w}(k+1)\|^2 + \|\mathbf{w}_* - \mathbf{w}_1\|^2 \\ &= \|\mathbf{w}_1 - \mathbf{w}'(k)\|^2 + \|\mathbf{w}_* - \mathbf{w}_1\|^2.\end{aligned}$$

It now only remains to show that $\|\mathbf{w}_* - \mathbf{w}_1\|^2 \leq \|\mathbf{w}_* - \mathbf{w}_0\|^2$. We have

$$\|\mathbf{w}_* - \mathbf{w}_0\|^2 = \frac{[d_* - \mathbf{w}^T(k)\mathbf{x}(k-1)]^2}{\|\mathbf{x}^\perp(k)\|^2} = \frac{e_1^2(k-1)}{\|\mathbf{x}^\perp(k)\|^2}$$

and

$$\|\mathbf{w}_* - \mathbf{w}_1\|^2 = \frac{[d_* - \mathbf{w}^T(k+1)\mathbf{x}(k-1)]^2}{\|\mathbf{x}^\perp(k)\|^2} = \frac{e_2^2(k-1)}{\|\mathbf{x}^\perp(k)\|^2}$$

where $e_1^2(k-1) \geq e_2^2(k-1)$ since $\mathbf{w}(k+1) \in \mathcal{H}(k-1)$.

Consequently we have $\|\mathbf{w}_* - \mathbf{w}(k+1)\|^2 \leq \|\mathbf{w}_* - \mathbf{w}(k)\|^2$. ■

Table 4.1: The Set-Membership Binormalized LMS Algorithm I

| SM-BNDRLMS-I Algorithm |
|--|
| <pre> for each k { $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { $\alpha(k) = 1 - \gamma/ e(k)$ $a = \mathbf{x}^T(k)\mathbf{x}(k)$ $\mathbf{w}'(k) = \mathbf{w}(k) + \alpha(k)e(k)\mathbf{x}(k)/a$ $\epsilon(k-1) = d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}'(k)$ if $\epsilon(k-1) > \gamma$ { $b = \mathbf{x}^T(k-1)\mathbf{x}(k-1)$ $c = \mathbf{x}^T(k)\mathbf{x}(k-1)$ $\beta(k) = 1 - \gamma/ \epsilon(k-1)$ $den = ab - c^2$ $\lambda_1 = -\epsilon(k-1)c/den$ $\lambda_2 = \epsilon(k-1)a/den$ $\mathbf{w}(k+1) = \mathbf{w}'(k) + \beta(k) [\lambda_1\mathbf{x}(k) + \lambda_2\mathbf{x}(k-1)]$ } } } else { $\mathbf{w}(k+1) = \mathbf{w}(k)$ } } </pre> |

4.3.2 Algorithm II

The SM-BNDRLMS-I algorithm in the previous subsection requires the intermediate check, that is if $\mathbf{w}'(k) \in \mathcal{H}(k)$, to determine if a second step is needed. This check will add extra computation. The algorithm proposed below, the SM-BNDRLMS-II, does not require this additional check to assure that $\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$. Let $\mathcal{S}_i(k)$ ($i = 1, 2$) denote the hyperplanes which contain all vectors \mathbf{w} such that $d(k-i+1) - \mathbf{w}^T\mathbf{x}(k-i+1) = g_i(k)$, where $g_i(k)$ are extra variables chosen such that the bound constraints are valid. That is, if $g_i(k)$ are chosen such that $|g_i(k)| \leq \gamma$, then $\mathcal{S}_i(k) \in \mathcal{H}(k-i+1)$.

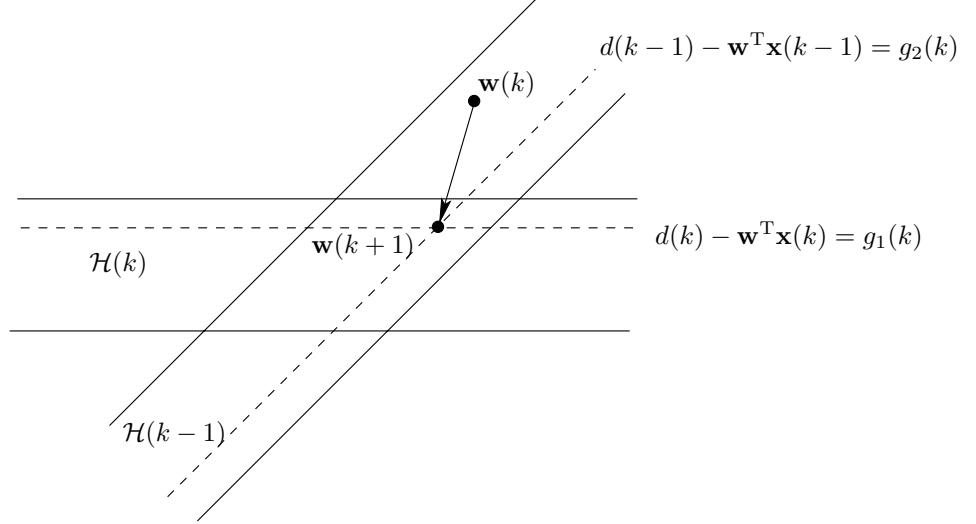


Figure 4.4: General algorithm update.

Consider the following optimization criterion whenever $\mathbf{w}(k) \notin \mathcal{H}(k) \cap \mathcal{H}(k-1)$

$$\mathbf{w}(k+1) = \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to:}$$

$$\begin{cases} d(k) - \mathbf{x}^T(k)\mathbf{w} & = g_1(k) \\ d(k-1) - \mathbf{x}^T(k-1)\mathbf{w} & = g_2(k) \end{cases} \quad (4.16)$$

The pair $\{g_1(k), g_2(k)\}$ specifies the point in $\mathcal{H}(k) \cap \mathcal{H}(k-1)$ where the final update will lie, see Figure 4.4. In order to evaluate if an update according to (4.16) is required, we need to first check if $\mathbf{w}(k) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$. Due to the concept of data reuse together with the constraint $|g_i(k)| \leq \gamma$, this check reduces to $\mathbf{w}(k) \in \mathcal{H}(k)$. Below we first solve for the general update, and thereafter consider a specific choice of the pair $\{g_1(k), g_2(k)\}$ leading to a simplified form.

To solve the optimization problem in (4.16), we can apply the method of Lagrange multipliers leading to the following objective function,

$$J_{\mathbf{w}} = \|\mathbf{w} - \mathbf{w}(k)\|^2 + \lambda_1[d(k) - \mathbf{x}^T(k)\mathbf{w} - g_1(k)]$$

$$+ \lambda_2[d(k-1) - \mathbf{x}^T(k-1)\mathbf{w} - g_2(k)] \quad (4.17)$$

After setting the gradient of (4.17) to zero and solving for the Lagrange multipliers, we get

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \frac{\lambda_1}{2}\mathbf{x}(k) + \frac{\lambda_2}{2}\mathbf{x}(k-1) & \text{if } |e(k)| > \gamma \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \quad (4.18)$$

where

$$\frac{\lambda_1}{2} = \frac{[e(k) - g_1(k)] \|\mathbf{x}(k-1)\|^2 - [\epsilon(k-1) - g_2(k)] \mathbf{x}^T(k)\mathbf{x}(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \quad (4.19)$$

$$\frac{\lambda_2}{2} = \frac{[\epsilon(k-1) - g_2(k)] \|\mathbf{x}(k)\|^2 - [e(k) - g_1(k)] \mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \quad (4.20)$$

where $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ and $\epsilon(k-1) = d(k-1) - \mathbf{w}^T(k)\mathbf{x}(k-1)$ are the *a priori* error at iteration k and the *a posteriori* error at iteration $k-1$, respectively.

Since $\mathbf{w}(k)$ always belongs to $\mathcal{H}(k-1)$ before a possible update we have $\epsilon(k-1) \leq \gamma$. Therefore choosing $g_2(k) = \epsilon(k-1)$ satisfies $|g_2(k)| \leq \gamma$. In the same way as in the SM-NLMS and SM-BNDRLMS-I algorithms, it is sufficient to choose $g_1(k)$ such that the update lies on closest boundary of $\mathcal{H}(k)$, i.e., $g_1(k) = \gamma e(k)/|e(k)|$. The above choices lead to the SM-BNDRLMS-II algorithm, where the new estimate $\mathbf{w}(k+1)$ will lie at the nearest boundary of $\mathcal{H}(k)$ such that the *a posteriori* error at iteration $k-1$, $\epsilon(k-1)$, is kept constant. A graphical illustration of the update procedure is shown in Figure 4.5. The update equations for the SM-BNDRLMS-II algorithm are given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda_1}{2}\mathbf{x}(k) + \frac{\lambda_2}{2}\mathbf{x}(k-1) \quad (4.21)$$

where

$$\begin{aligned} \frac{\lambda_1}{2} &= \frac{\alpha(k)e(k)\|\mathbf{x}(k-1)\|^2}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \\ \frac{\lambda_2}{2} &= -\frac{\alpha(k)e(k)\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \\ \alpha(k) &= \begin{cases} 1 - \frac{\gamma}{|e(k)|}, & \text{if } |e(k)| > \gamma \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (4.22)$$

As with the SM-BNDRLMS-I algorithm in the previous subsection, the problem with parallel constraint sets is avoided by using the SM-NLMS update of (4.5) whenever the denominator in the λ_i is zero. Table 4.2 summarizes the recursions of the SM-BNDRLMS-II algorithm.

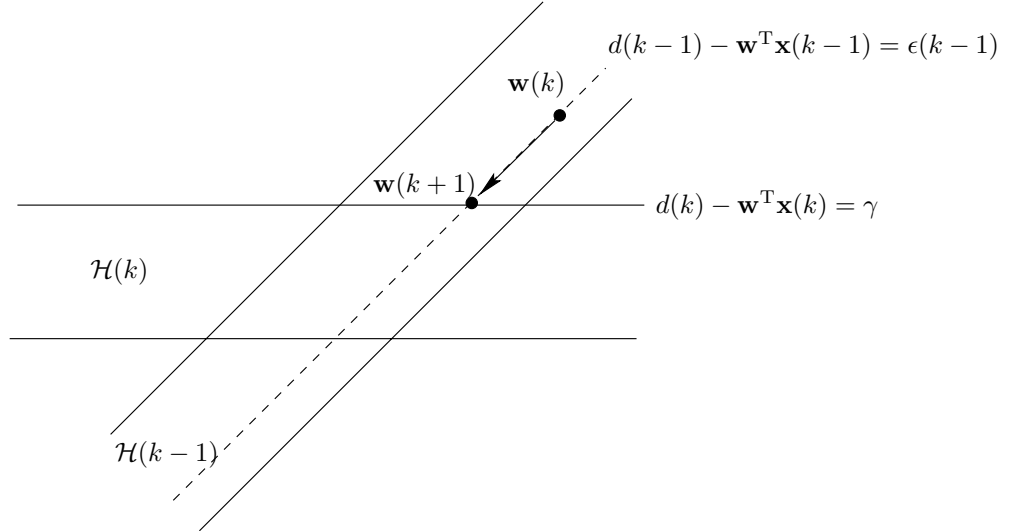


Figure 4.5: The SM-BNDRLMS-II algorithm.

4.3.3 Computational Complexity

The computational complexity per update in terms of the number of additions, multiplications, and divisions for the three algorithms are shown in Table 4.3. For the SM-BNDRLMS-I, the two possible update complexities are listed where the first corresponds to the total complexity when only the first step is necessary, i.e., when $\mathbf{w}'(k) \in \mathcal{H}(k-1)$, and the second corresponds to the total complexity when a full update is needed. Applying the SM-BNDRLMS algorithms slightly increases the computational complexity as compared with that of the SM-NLMS algorithm. However, the SM-BNDRLMS algorithms have a reduced number of updates and an increased convergence rate as compared to the SM-NLMS algorithm, as verified through simulations in Section 4.5. Comparing the complexities of the SM-BNDRLMS-I and SM-BNDRLMS-II algorithms, we note that the difference in the overall complexity depends on the frequency the second step is required in Algorithm I. In the operation counts, the value of $\|\mathbf{x}(k-1)\|^2$ at iteration k was assumed unknown. However, once $\|\mathbf{x}(k)\|^2$ or $\|\mathbf{x}(k-1)\|^2$ is known one can compute the other using only two additional multiplications, e.g., $\|\mathbf{x}(k-1)\|^2 = \|\mathbf{x}(k)\|^2 - x^2(k) + x(k-N)^2$. The relation between $\|\mathbf{x}(k-1)\|^2$ and $\|\mathbf{x}(k)\|^2$ has been used in the operation counts of the SM-BNDRLMS algorithms. If an update occurs at two successive time instants, $\|\mathbf{x}(k-1)\|^2$

Table 4.2: The Set-Membership Binormalized LMS Algorithm II

| SM-BNDRLMS-II Algorithm |
|--|
| <pre> for each k { $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { $\alpha(k) = 1 - \gamma/ e(k)$ $a = \mathbf{x}^T(k)\mathbf{x}(k)$ $b = \mathbf{x}^T(k-1)\mathbf{x}(k-1)$ $c = \mathbf{x}^T(k)\mathbf{x}(k-1)$ $den = ab - c^2$ $\lambda_1 = e(k)b/den$ $\lambda_2 = -e(k)c/den$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k) [\lambda_1\mathbf{x}(k) + \lambda_2\mathbf{x}(k-1)]$ } else { $\mathbf{w}(k+1) = \mathbf{w}(k)$ } } </pre> |

and $\mathbf{x}^T(k-1)\mathbf{x}(k-2)$ are known from previous update, and as a consequence the number of multiplications and additions in such updates can be further reduced by approximately N for the SM-NLMS algorithm and $2N$ for the SM-BNDRLMS algorithms. Finally, note that if we continuously estimate $\|\mathbf{x}(k)\|^2$ and $\mathbf{x}^T(k)\mathbf{x}(k-1)$, regardless if an update is required or not, the SM-BNDRLMS-II algorithm will always be more efficient than SM-BNDRLMS-I. These savings in computations are crucial in applications where the filter order is high and computational resources are limited.

Table 4.3: Computational Complexity per Update

| ALG. | MULT. | ADD. | DIV. |
|------------------------|----------|----------|------|
| SM-NLMS | $3N + 1$ | $3N$ | 2 |
| SM-BNDRLMS-I (1 step) | $4N + 1$ | $4N$ | 2 |
| SM-BNDRLMS-I (2 steps) | $7N + 8$ | $7N + 3$ | 4 |
| SM-BNDRLMS-II | $5N + 7$ | $5N + 3$ | 2 |

4.4 Second-Order Statistical Analysis

This section addresses the steady-state analysis of the SM-BNDRLMS algorithms.

4.4.1 Coefficient-Error Vector

In this subsection we investigate the convergence behavior of the coefficient vector $\mathbf{w}(k)$. It is assumed that an unknown FIR \mathbf{w}_{opt} is identified with an adaptive filter $\mathbf{w}(k)$ of the same order $N - 1$ using the SM-BNDRLMS-II algorithm. The desired response is given by

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt} + n(k) \quad (4.23)$$

where $n(k)$ is measurement noise, assumed here to be Gaussian with zero mean and variance σ_n^2 . We study the evolution of the coefficient error $\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$. The output error can now be written as

$$e(k) = n(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k) \quad (4.24)$$

The update equations for the adaptive filter coefficients are given by

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) & \text{if } |e(k)| \leq \gamma \\ \mathbf{w}(k) + [e(k) - \gamma]\mathbf{a} & \text{if } e(k) > +\gamma \\ \mathbf{w}(k) + [e(k) + \gamma]\mathbf{a} & \text{if } e(k) < -\gamma \end{cases} \quad (4.25)$$

where

$$\mathbf{a} = \frac{\|\mathbf{x}(k-1)\|^2\mathbf{x}(k) - [\mathbf{x}^T(k-1)\mathbf{x}(k)]\mathbf{x}(k-1)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (4.26)$$

As a consequence, the coefficient error at time instant $k+1$ becomes

$$\Delta\mathbf{w}(k+1) = \begin{cases} \Delta\mathbf{w}(k) & \text{if } |e(k)| \leq \gamma \\ [\mathbf{I} + \mathbf{A}]\Delta\mathbf{w}(k) + \mathbf{b} - \mathbf{c} & \text{if } e(k) > +\gamma \\ [\mathbf{I} + \mathbf{A}]\Delta\mathbf{w}(k) + \mathbf{b} + \mathbf{c} & \text{if } e(k) < -\gamma \end{cases} \quad (4.27)$$

where

$$\mathbf{A} = \frac{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k) - \|\mathbf{x}(k-1)\|^2\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (4.28)$$

and

$$\begin{aligned}\mathbf{b} &= n(k)\mathbf{a} \\ \mathbf{c} &= \gamma\mathbf{a}\end{aligned}\tag{4.29}$$

In the analysis, we utilize the following initial assumptions:

AS1) The filter is updated with the probability $P_e(k) = Pr\{|e(k)| > \gamma\}$, and $Pr\{e(k) > \gamma\} = Pr\{e(k) < -\gamma\}$.

Note that the probability $P_e(k)$ will be time-varying because the variance of the output error, $e(k)$, depends on the mean of the squared coefficient-error vector norm and for Gaussian noise with zero mean and variance σ_n^2 we get $\sigma_e^2 = \sigma_n^2 + E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)]$. Since we are interested in the excess MSE we will assume hereafter that

AS2) Since we are only interested in the excess MSE we will hereafter assume that the filter has reached the steady-state value.

4.4.2 Input-Signal Model

In the evaluation of the excess MSE we use a simplified model for the input signal vector $\mathbf{x}(k)$. The model uses a simplified distribution for the input-signal vector by employing reduced and countable angular orientations for the excitation, which are consistent with the first- and second-order statistics of the actual input-signal vector. The model was used for analyzing the NLMS algorithm [44] as well as the BNDRLMS algorithm [51], and was shown to yield good results.

The input signal vector for the model is

$$\mathbf{x}(k) = s(k)r(k)\mathbf{v}(k)\tag{4.30}$$

where

- $s(k)$ is ± 1 with probability $1/2$

- $r^2(k)$ has the same probability distribution as $\|\mathbf{x}(k)\|^2$, and in the case of white Gaussian input signal it is a sample of an independent process with χ -square distribution with N degrees of freedom, with $E[r^2(k)] = N\sigma_x^2$
- $\mathbf{v}(k)$ is one of the N orthonormal eigenvectors of $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$, say $\{\mathcal{V}_i, i = 1, \dots, N\}$. For a white Gaussian input signal, it is assumed that $\mathbf{v}(k)$ is uniformly distributed such that

$$Pr\{\mathbf{v}(k) = \mathcal{V}_i\} = \frac{1}{N} \quad (4.31)$$

4.4.3 Excess MSE for White Input Signals

In this subsection we investigate the excess MSE in the SM-BNDRLMS algorithms. In order to achieve this goal we have to consider a simple model for the input signal vector which assumes a discrete set of angular orientations. The excess MSE is given by [1]

$$\xi_{exc} = \lim_{k \rightarrow \infty} \xi(k) - \xi_{min} \quad (4.32)$$

where

$$\xi(k) = E[e^2(k)] = E\left[\{n(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k)\}^2\right] \quad (4.33)$$

is the MSE at iteration k and ξ_{min} is the minimum MSE. With these equations, we have

$$\begin{aligned} \Delta\xi(k) &= E\left[\{n(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k)\}^2\right] - \xi_{min} \\ &= E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \\ &= \text{tr}\{\mathbf{R}_{cov}[\Delta\mathbf{w}(k)]\} \end{aligned} \quad (4.34)$$

For the input-signal model presented in the previous subsection, $\Delta\xi(k+1)$ can be written as

$$\begin{aligned} \Delta\xi(k+1) &= \Delta\xi(k+1)|_{\mathbf{x}(k)\parallel\mathbf{x}(k-1)} \times P[\mathbf{x}(k)\parallel\mathbf{x}(k-1)] \\ &\quad + \Delta\xi(k+1)|_{\mathbf{x}(k)\perp\mathbf{x}(k-1)} \times P[\mathbf{x}(k)\perp\mathbf{x}(k-1)] \end{aligned} \quad (4.35)$$

Conditions $\mathbf{x}(k)\parallel\mathbf{x}(k-1)$ and $\mathbf{x}(k)\perp\mathbf{x}(k-1)$ in the model are equivalent to $\mathbf{v}(k) = \mathbf{v}(k-1)$ and $\mathbf{v}(k) \neq \mathbf{v}(k-1)$, respectively, because $\mathbf{v}(k)$ and $\mathbf{v}(k-1)$ can only be parallel or

orthogonal to each other. $P[\mathbf{x}(k)\|\mathbf{x}(k-1)]$ denotes the probability that $\mathbf{x}(k)\|\mathbf{x}(k-1)$, and $P[\mathbf{x}(k)\perp\mathbf{x}(k-1)]$ the denotes probability that $\mathbf{x}(k)\perp\mathbf{x}(k-1)$. For the case $\mathbf{x}(k)\|\mathbf{x}(k-1)$, the SM-BNDRLMS algorithm will behave like the SM-NLMS algorithm which has the excess MSE (see Appendix A4.1 of this chapter)

$$\Delta\xi(k+1)|_{\mathbf{x}(k)\|\mathbf{x}(k-1)} = \left(1 - \frac{2P_e(k) - P_e^2(k)}{N}\right) \Delta\xi(k) + P_e^2(k) \frac{\sigma_n^2}{N+1-\nu_x} \quad (4.36)$$

where $\nu_x = E[x^4(k)/\sigma_x^4]$ varies from 1 for binary distribution, to 3 for Gaussian distribution, to ∞ for a Cauchy distribution [52, 44]. For the case $\mathbf{x}(k)\perp\mathbf{x}(k-1)$ the expression for the coefficient error vector also reduces to the same as that of the SM-NLMS algorithm (see Appendix A4.2 of this chapter) giving

$$\Delta\xi(k+1)|_{\mathbf{x}(k)\perp\mathbf{x}(k-1)} = \left(1 - \frac{2P_e(k) - P_e^2(k)}{N}\right) \Delta\xi(k) + P_e^2(k) \frac{\sigma_n^2}{N+1-\nu_x} \quad (4.37)$$

Combining we have

$$\begin{aligned} \Delta\xi(k+1) &= \Delta\xi(k+1)|_{\mathbf{x}(k)\|\mathbf{x}(k-1)} P[\mathbf{x}(k)\|\mathbf{x}(k-1)] \\ &\quad + \Delta\xi(k+1)|_{\mathbf{x}(k)\perp\mathbf{x}(k-1)} P[\mathbf{x}(k)\perp\mathbf{x}(k-1)] \\ &= (P[\mathbf{x}(k)\|\mathbf{x}(k-1)] + P[\mathbf{x}(k)\perp\mathbf{x}(k-1)]) \Delta\xi^{\parallel}(k) \\ &= (P[\mathbf{x}(k)\|\mathbf{x}(k-1)] + P[\mathbf{x}(k)\perp\mathbf{x}(k-1)]) \Delta\xi^{\perp}(k) \\ &= \left(1 - \frac{2P_e(k) - P_e^2(k)}{N}\right) \Delta\xi(k) + P_e^2(k) \frac{\sigma_n^2}{N+1-\nu_x} \end{aligned} \quad (4.38)$$

Recall assumption AS2) of the filter to being in steady-state such that the probability $P_e(k) \rightarrow P_e$ is constant. The stability and convergence of (4.38) holds since $P_e(k) \leq 1$. If we let $k \rightarrow \infty$, the excess MSE becomes

$$\xi_{exc} = \frac{N}{N+1-\nu_x} \cdot \frac{P_e \sigma_n^2}{2-P_e} \quad (4.39)$$

Assuming the filter has converged to its steady-state value, the probability of update for white Gaussian input signals is given by

$$P_e = 2Q\left(\frac{\gamma}{\sqrt{\sigma_n^2 + \sigma_x^2 E[\|\Delta\mathbf{w}_{\infty}\|^2]}}\right) \quad (4.40)$$

where $Q(\cdot)$ is the complementary Gaussian cumulative distribution function given by

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt \quad (4.41)$$

and $\mathbb{E}[\|\Delta\mathbf{w}_\infty\|^2]$ is the mean of the squared norm of the coefficient error after convergence. To be able to calculate the expression in (4.39) we need P_e which in turn depends on $\sigma_x^2 \mathbb{E}[\|\Delta\mathbf{w}_\infty\|^2]$. Therefore consider the following two cases of approximation:

AP1) The variance of the error is lower bounded by the noise variance, i.e., $\sigma_e^2 = \sigma_n^2 + \sigma_x^2 \mathbb{E}[\|\Delta\mathbf{w}_\infty\|^2] \geq \sigma_n^2$. Therefore, a simple lower bound is given by $\hat{P}_e \geq 2Q\left(\frac{\gamma}{\sigma_n}\right)$

AP2) We can rewrite the variance of the error as $\sigma_{e(k)}^2 = \sigma_n^2 + \mathbb{E}[\tilde{e}^2(k)]$, where $\tilde{e}(k) = e(k) - e_{opt}$ denotes the distance between the error at k th iteration and the optimal error. Assuming no update we have $|e(k)| \leq \gamma$, and with $\sigma_{opt}^2 = \sigma_n^2$ we get $\sigma_{e(k)}^2 \leq 2\sigma_n^2 + \gamma^2$. Therefore, an upper bound of the probability of update is given by $\hat{P}_e = 2Q\left(\frac{\gamma}{\sigma_e}\right) \leq 2Q\left(\frac{\gamma}{\sqrt{2\sigma_n^2 + \gamma^2}}\right)$

The approximations of P_e together with Equation (4.39) are used in the simulations to estimate the excess MSE for different thresholds γ .

4.4.4 Excess MSE for Colored Input Signals

When extending the analysis to colored input signals we may still use the input-signal model in (4.30). The angular distributions of $\mathbf{x}(k)$ will change, i.e., the probabilities $P[\mathbf{x}(k) \parallel \mathbf{x}(k-1)]$ and $P[\mathbf{x}(k) \perp \mathbf{x}(k-1)]$ will be different from those for white input signals. However, as with the case of white input signals, these probabilities will not have effect on the final results, see Equation (4.38). In order to get an expression for the probability of update P_e for colored input signals we assume that the input is correlated according to

$$x(k) = rx(k-1) + (1-r)v(k) \quad (4.42)$$

where $v(k)$ is a white noise process with zero mean and variance σ_v^2 . Straightforward calculations give the autocorrelation matrix

$$\mathbf{R} = \sigma_x^2 \begin{bmatrix} 1 & r & r^2 & \dots & r^{N-1} \\ r & 1 & r & \dots & r^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r^{N-1} & r^{N-2} & r^{N-3} & \dots & 1 \end{bmatrix} \quad (4.43)$$

where

$$\sigma_x^2 = \frac{1-r}{1+r} \sigma_v^2 = b \sigma_v^2 \quad (4.44)$$

Assuming the filter has converged to its steady-state, the variance of the output error can now be computed as

$$\begin{aligned} \sigma_e^2 &= \sigma_n^2 + \mathbb{E}[\Delta \mathbf{w}_\infty^T \mathbf{R} \Delta \mathbf{w}_\infty] \\ &\leq \sigma_n^2 + \frac{\sigma_x^2}{b} \mathbb{E}[\|\Delta \mathbf{w}_\infty\|^2] \end{aligned} \quad (4.45)$$

where the last inequality is shown in Appendix 4.3. The probability of update is now given by

$$P_e \leq Q \left(\frac{\gamma}{\sqrt{\sigma_n^2 + b^{-1} \sigma_x^2 \mathbb{E}[\|\Delta \mathbf{w}_\infty\|^2]}} \right) \quad (4.46)$$

To be able to evaluate the probability of update P_e the same approximation is made as in AP2) for the case of white input signals, i.e., $\sigma_x^2 \mathbb{E}[\|\Delta \mathbf{w}_\infty\|^2] \leq \sigma_n^2 + \gamma^2$. An upper bound for the case of colored input signals is now given by $\hat{P}_e \leq 2Q \left(\frac{\gamma}{\sqrt{(1+b^{-1})\sigma_n^2 + b^{-1}\gamma^2}} \right)$. The lower bound given in AP1) in the previous section is still valid.

4.5 Simulation Results

In this section, the new algorithms are applied to a system identification problem. The order of the plant was $p = N - 1 = 10$ and the input signal was colored noise with condition number 100. The signal-to-noise ratio (SNR) was set to 80dB and 20dB in two different examples.

Figure 4.6 shows the learning curves averaged over 500 simulations for the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS, the BNDRLMS, and the NLMS algorithms for an $SNR = 80$ dB. The upper bound on the estimation error was set to $\gamma = \sqrt{5}\sigma_n$, and the step sizes used in the BNDRLMS and the NLMS algorithms were set to unity in order to obtain the fastest convergence.

Figure 4.6 clearly shows how the SM-BNDRLMS-I, and the SM-BNDRLMS-II algorithms combine the fast convergence of the BNDRLMS algorithm with the low misadjustment of the SM-NLMS algorithm. In an ensemble of 500 experiments of 1000 iterations the average number of updates per experiment for the SM-BNDRLMS-I, SM-BNDRLMS-II, and the SM-NLMS algorithms were, 185, 180, and 436 respectively. For the SM-BNDRLMS-I an average of 108 updates were full updates.

Figure 4.7 shows the learning curves results for an $SNR = 20$ dB. The parameters used in the algorithms were the same as in the first example. As can be seen from the figure, the SM-BNDRLMS algorithms still have higher convergence speeds than the SM-NLMS algorithm.

In 1000 iterations, the average number of updates per experiment for the SM-BNDRLMS-I, SM-BNDRLMS-II, and the SM-NLMS algorithms were, 100, 95, and 129 respectively. For the SM-BNDRLMS-I an average of 15 updates were full updates.

In the two examples above the NLMS and the BNDRLMS algorithms were unable to reach the same low steady-state value as their set-membership versions, and a trade-off between convergence speed and final MSE was observed.

For the two examples above we also plotted the overall complexity versus the total number of iterations for the SM-NLMS and the SM-BNDRLMS algorithms. The curves are normalized with respect to the number of filter coefficients N . To minimize the computational complexity for all the algorithms, we recursively estimated $\|\mathbf{x}(k)\|^2$ and $\mathbf{x}^T(k)\mathbf{x}(k-1)$ at each iteration. Figures 4.8 and 4.9 show the results based on the above simulations. For the case of high SNR, we see from Figure 4.8 that the overall complexity of the SM-BNDRLMS algorithms are initially higher than the SM-NLMS algorithm. As time proceeds the overall complexity of the SM-BNDRLMS-II algorithm becomes similar to that of the SM-NLMS

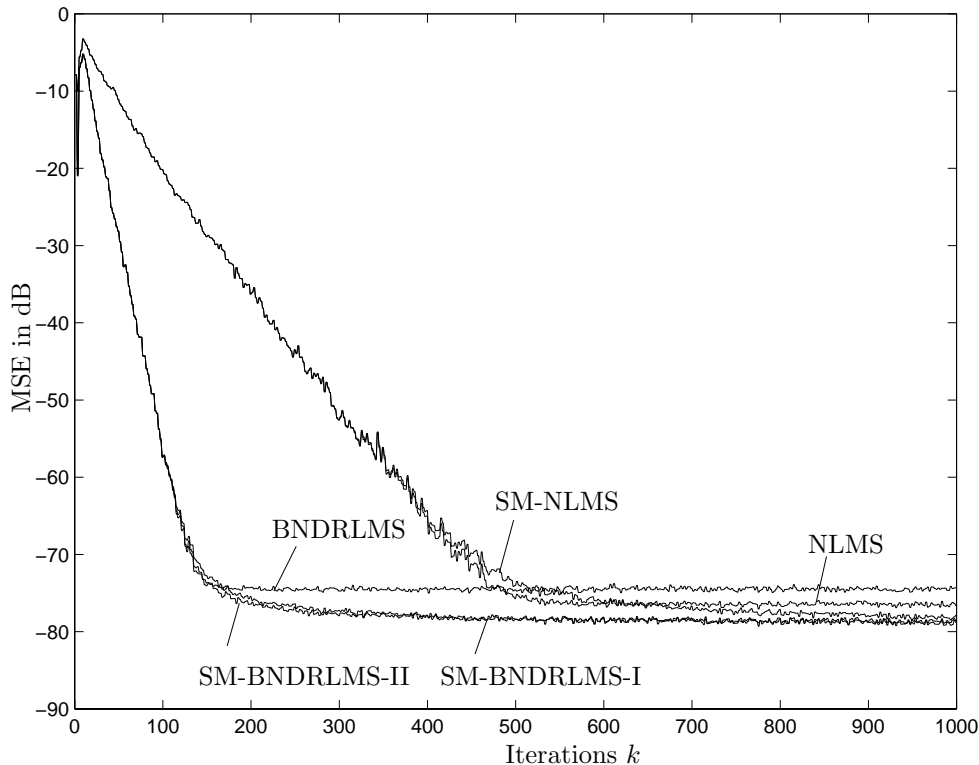


Figure 4.6: Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS, the BNDRLMS, and the NLMS algorithms. Condition number of the input-signal correlation matrix = 100, SNR = 80 dB, and $\gamma = \sqrt{5}\sigma_n$.

algorithm. The SM-BNDRLMS-II, with its extra innovation check, tends to a slightly higher value. For a low SNR the SM-NLMS algorithm will have a slightly lower overall complexity as compared to the SM-BNDRLMS algorithms.

The algorithms were also tested for a low SNR of 10 dB. The adaptive filter had $N = 60$ coefficients. The learning curves for the SM-NLMS and SM-BNDRLMS algorithms are shown in Figure 4.10. If the order of the filter is decreased, the differences between convergence rate of the algorithms decrease. This confirms the intuition that for high background noise, the simplest algorithms could provide as good performance as more dedicated ones.

In order to test the algorithms in a time-varying environment, the system coefficients were changed according to the model, $\mathbf{w}_{opt}(k) = \mathbf{w}_{opt}(k-1) + \mathbf{u}(k)$, where $\mathbf{u}(k)$ is a random vector with elements of zero mean and variance $\sigma_v^2 = 10^{-6}$. In the simulations the additive

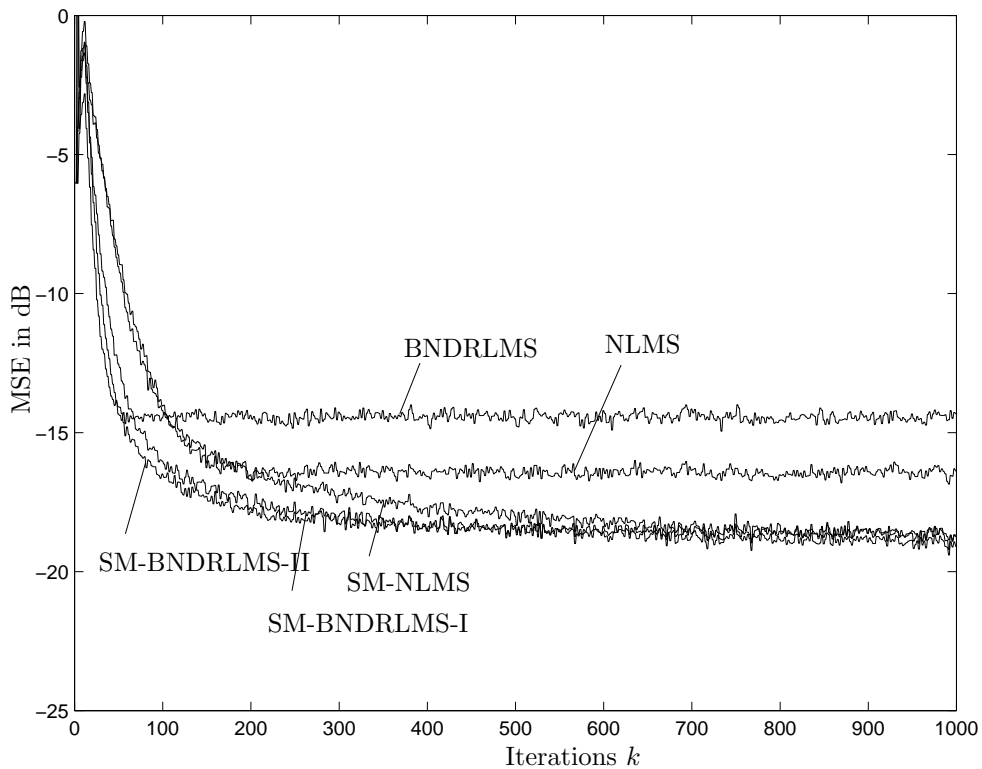


Figure 4.7: Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS, the BNDRLMS, and the NLMS algorithms. Condition number of the input-signal correlation matrix = 100, SNR = 20 dB, and $\gamma = \sqrt{5}\sigma_n$.

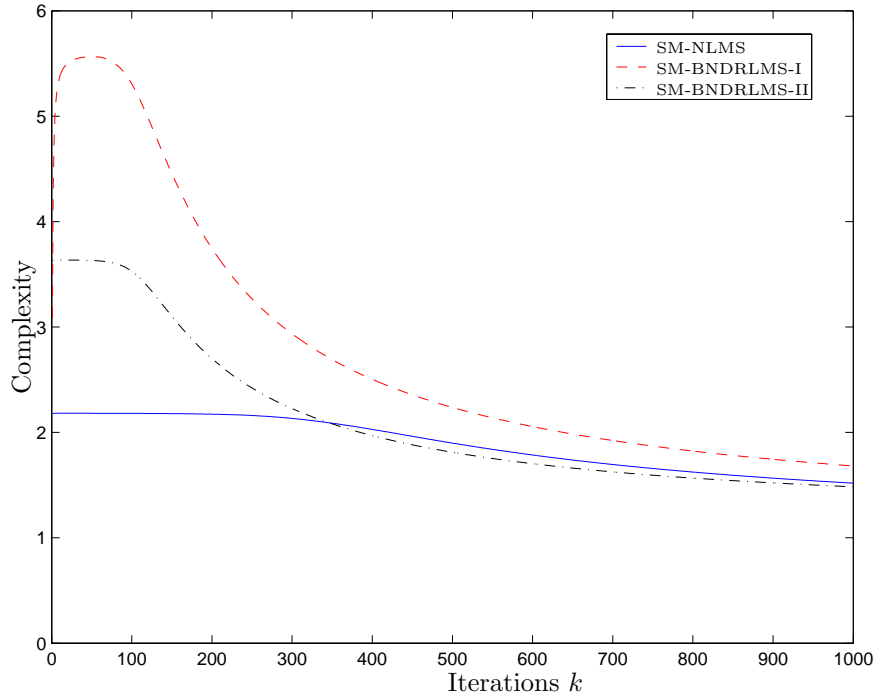


Figure 4.8: The overall complexity normalized with N versus the number of data points in the simulation for $SNR = 80$ dB.

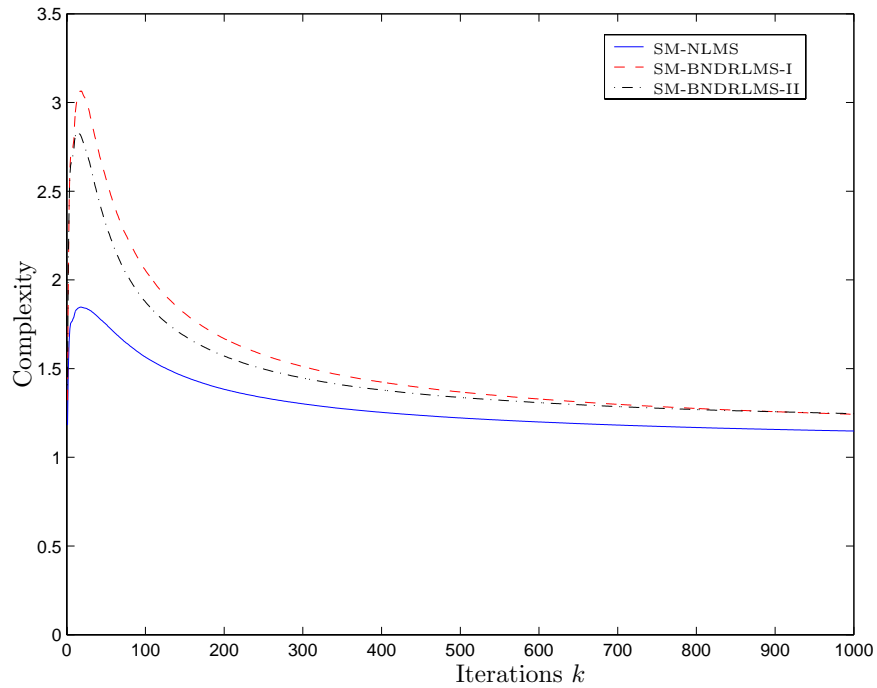


Figure 4.9: The overall complexity normalized with N versus the number of data points in the simulation for $SNR = 20$ dB.

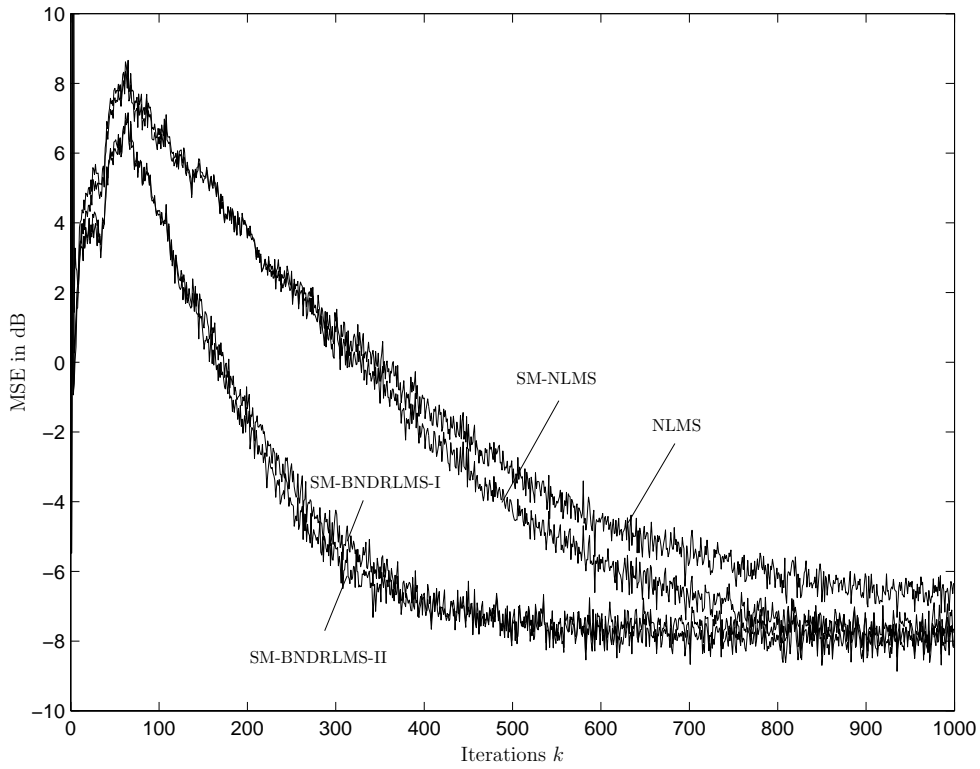


Figure 4.10: Learning curves of the SM-BNDRLMS-I, the SM-BNDRLMS-II, the SM-NLMS. Condition number of the input-signal correlation matrix = 100, SNR = 10 dB, and $\gamma = \sigma_n$.

Table 4.4: Excess Mean-Square Error in Nonstationary Environments

| ALG. | ξ_{exc} (dB) |
|---------------|-------------------------|
| NLMS | -40.8 |
| BNDRLMS | -43.5 |
| SM-NLMS | -40.8 |
| SM-BNDRLMS-I | -43.4 |
| SM-BNDRLMS-II | -43.5 |

noise was set to zero, and the bound on the estimation error was set to $\gamma = \sqrt{5}\sigma_v$. The results in terms of the excess MSE in dB can be found in Table 4.4. As can be noticed the new proposed algorithms present tracking performance comparable to the BNDRLMS algorithm.

Finally experiments were conducted to validate the theoretical results obtained in the MSE analysis. The MSE was measured for different values of γ (γ varied from σ_n to $\sqrt{10}\sigma_n$). The order of the plant was $N - 1 = 10$, and the SNR was chosen to 60 dB.

Figure 4.11 shows the MSE versus γ^2/σ_n^2 for a modeled input signal, where the input vectors were chosen such that $\mathbf{v}(k)$ and $\mathbf{v}(k - 1)$ were parallel or orthogonal with probabilities $\frac{1}{N}$ and $\frac{N-1}{N}$, respectively. As can be seen from the figure, the theoretical curves can predict the behavior of the simulation for the assumed model. Figures 4.12 and 4.13 show the results for white and colored input signals, respectively. In the case of colored input, the condition number of the input-signal correlation matrix was equal to 100. It was shown in [69] that the output error $e(k)$ is upper bounded by γ after that convergence has taken place. Therefore, we can conclude that the MSE is upper bounded by γ^2 . However, from the figures it can be seen that the theoretical formulas for the MSE can provide a much tighter bound than simply considering $\sigma_e^2 = \gamma^2$. If we use this upper bound in AP1) together with Equation (4.39), the difference for the white input case will be between 2.5 dB and 10 dB for γ^2/σ^2 in the range 2–10.

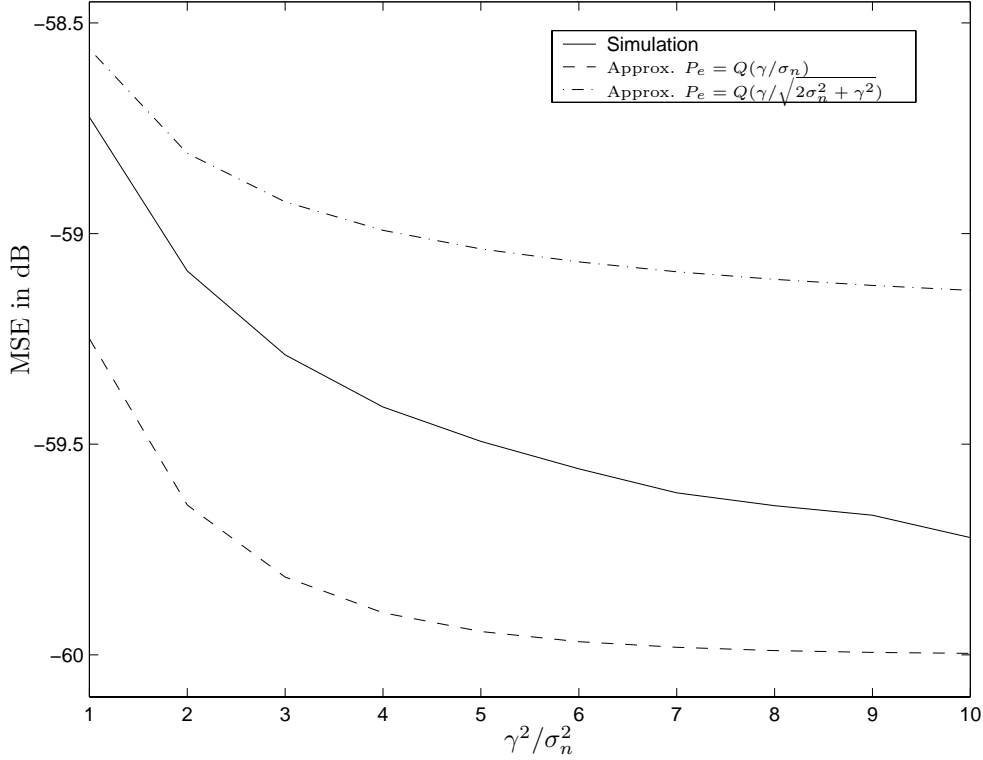


Figure 4.11: MSE for $N = 10$ as function of γ^2/σ_n^2 , for the input signals as modeled.

4.6 Conclusions

This chapter derived two novel adaptation algorithms based on the concept of set-membership filtering. The algorithms utilize consecutive data-pairs in order to construct a space of feasible solutions for the updates. The new algorithms were applied to a system identification problem, in order to verify the good performance of the algorithm when compared with the SM-NLMS algorithm in terms of high convergence speed, low misadjustment, and reduced number of updates. Analysis for the mean-squared error was carried out for both white and colored input signals, and closed form expression for the excess MSE was provided.

Appendix A4.1

For the special case that $\mathbf{x}(k) \parallel \mathbf{x}(k-1)$ the recursions of the SM-BNDRLMS algorithm will be equal to those of the SM-NLMS algorithm. In the derivations below $\mathbf{x}(k)$ is replaced by $s(k)r(k)\mathbf{v}(k)$, and the second-order approximation $E[1/r^2(k)] \approx 1/[\sigma_x^2(N+1-\nu_x)]$

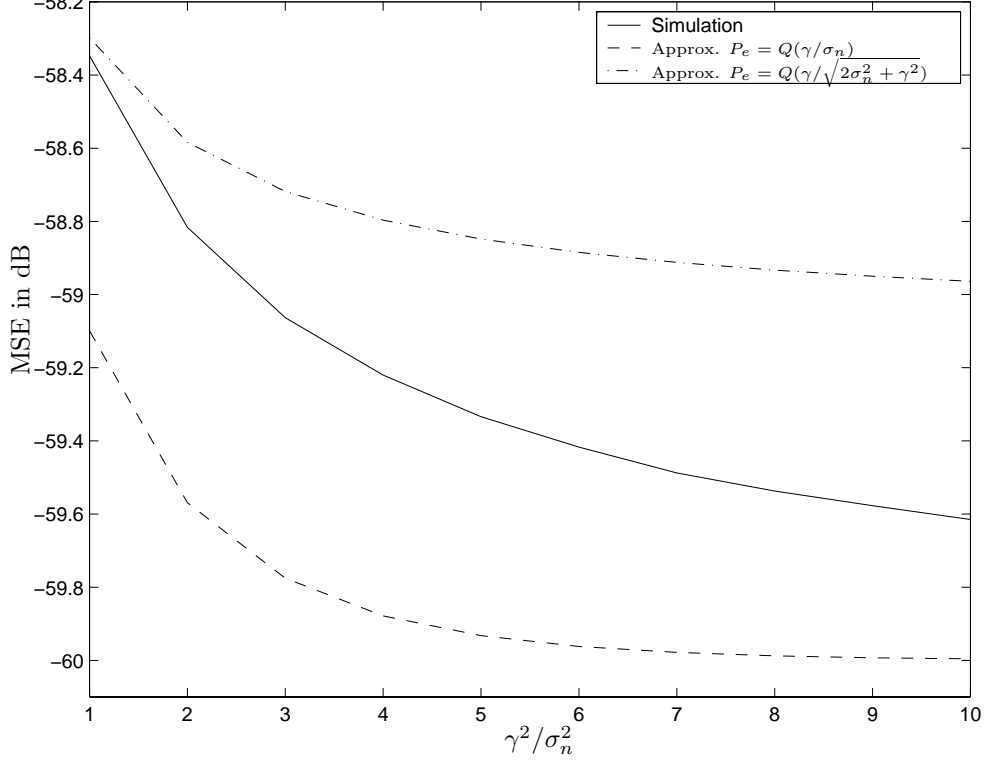


Figure 4.12: MSE for $N = 10$ as function of γ^2/σ_n^2 for white input signals.

introduced in [44] is used. The coefficient error at time instant $k + 1$ expressed in terms of the probability $P_e(k)$ can be easily derived in the same manner as with the SM-BNDRLMS algorithms in Section 4.4 and is given by

$$\Delta \mathbf{w}(k+1) = \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \Delta \mathbf{w}(k) + P_e(k) \frac{n(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \quad (4.47)$$

For white input signal we have $\mathbf{R} = \sigma_x^2 \mathbf{I}$. The expression for $\Delta \xi(k+1)$ is given by

$$\begin{aligned} \Delta \xi(k+1) &= \sigma_x^2 \text{tr}(\text{cov}[\Delta \mathbf{w}(k+1)]) = \sigma_x^2 \text{tr}(\mathbf{E}[\Delta \mathbf{w}(k+1)\Delta \mathbf{w}^T(k+1)]) \\ &= \sigma_x^2 \text{tr} \left(\mathbf{E} \left\{ \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \Delta \mathbf{w}(k)\Delta \mathbf{w}^T(k) \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} \right) \\ &\quad + \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbf{E} \left\{ \frac{n^2(k)\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right\} \right) \\ &= \psi_1 + \psi_2 \end{aligned} \quad (4.48)$$

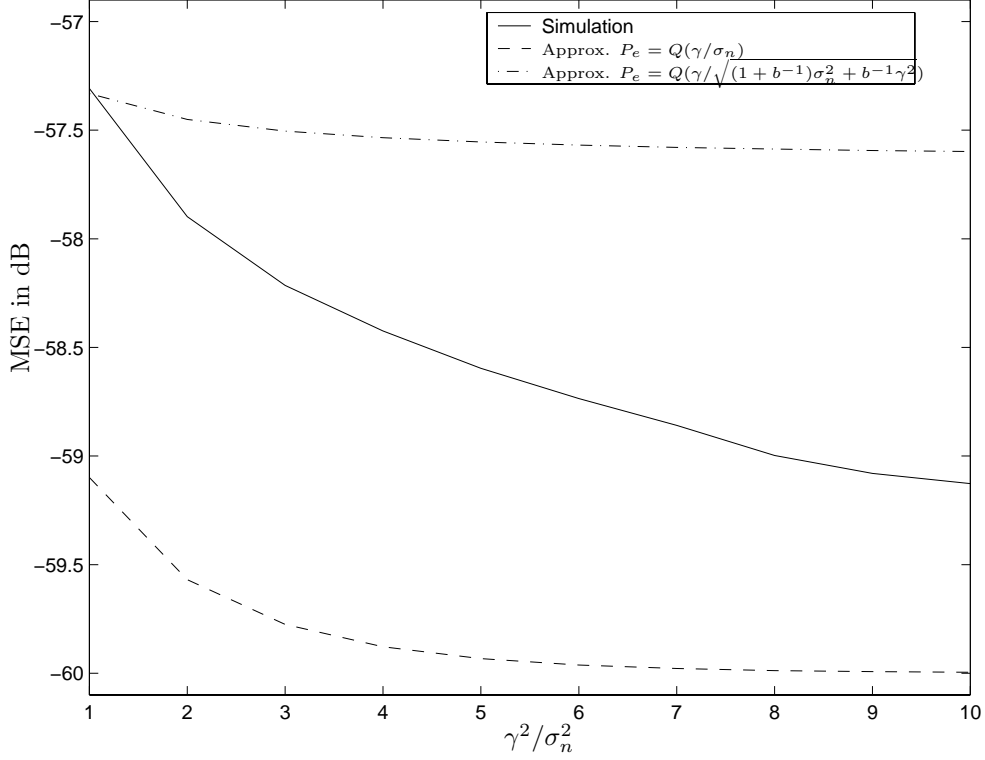


Figure 4.13: MSE for $N = 10$ as function of γ^2/σ_n^2 for colored input signals.

where

$$\begin{aligned}
\psi_1 &= \sigma_x^2 \text{tr} \left(\mathbb{E} \left\{ \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right\} \right) \\
&= \sigma_x^2 \text{tr} \left(\mathbb{E} \{ \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \} \right) \\
&\quad - \sigma_x^2 P_e(k) \text{tr} \left(\mathbb{E} \left[\frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right) \\
&\quad - \sigma_x^2 P_e(k) \text{tr} \left(\mathbb{E} \left[\frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right) \\
&\quad + \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbb{E} \left[\frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right) \\
&= \rho_1 + \rho_2 + \rho_3 + \rho_4 \tag{4.49}
\end{aligned}$$

with

$$\rho_1 = \sigma_x^2 \text{tr} \left(\mathbb{E} \left[\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \right] \right) = \Delta \xi(k) \tag{4.50}$$

$$\begin{aligned}
\rho_2 &= -\sigma_x^2 P_e(k) \text{tr} \left(\mathbb{E} \left[\frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \right) \\
&= -\sigma_x^2 P_e(k) \text{tr} \left(\mathbb{E} \left[\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{v}(k) \mathbf{v}^T(k) \right] \right) \\
&= -\frac{\sigma_x^2}{N} P_e(k) \text{tr} \left(\mathbb{E} \left[\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \right] \right) \\
&= -\frac{P_e(k)}{N} \Delta \xi(k) \\
&= \rho_3
\end{aligned} \tag{4.51}$$

where the last equality is true since $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$.

$$\begin{aligned}
\rho_4 &= \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbb{E} \left[\mathbf{A} \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{A} \right] \right) \\
&= \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbb{E} \left[\frac{\mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k)}{[\|\mathbf{x}(k)\|^2]^2} \right] \right) \\
&= \sigma_x^2 P_e^2(k) \mathbb{E} \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{[\|\mathbf{x}(k)\|^2]^2} \right] \\
&= \sigma_x^2 P_e^2(k) \mathbb{E} \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{x}(k)\|^2} \right] \\
&= \sigma_x^2 P_e^2(k) \mathbb{E} \left[\Delta \mathbf{w}^T(k) \mathbf{v}(k) \mathbf{v}^T(k) \Delta \mathbf{w}(k) \right] \\
&= \frac{\sigma_x^2 P_e^2(k)}{N} \text{tr} \left(\mathbb{E} \left[\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \right] \right) \\
&= \frac{P_e^2(k)}{N} \Delta \xi(k)
\end{aligned} \tag{4.52}$$

$$\begin{aligned}
\psi_2 &= \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbb{E} \left\{ \frac{n^2(k) \mathbf{x}(k) \mathbf{x}^T(k)}{[\|\mathbf{x}(k)\|^2]^2} \right\} \right) \\
&= \sigma_x^2 P_e^2(k) \text{tr} \left(\mathbb{E} \left[n^2(k) \frac{\mathbf{v}(k) \mathbf{v}^T(k)}{r^2} \right] \right) \\
&= \sigma_x^2 P_e^2(k) \sigma_n^2 \left(\mathbb{E} \left[\frac{1}{r^2} \right] \right) \\
&= \frac{\sigma_n^2 P_e^2(k)}{N + 1 - \nu_x}
\end{aligned} \tag{4.53}$$

Finally, we get

$$\Delta \xi(k+1) |_{\mathbf{x}(k) \|\mathbf{x}(k-1)} = \left(1 - \frac{2P_e(k) - P_e^2(k)}{N} \right) \Delta \xi(k) + P_e^2(k) \frac{\sigma_n^2}{N + 1 - \nu_x} \tag{4.54}$$

Appendix A4.2

For the case $\mathbf{x}(k) \perp \mathbf{x}(k-1)$, (4.28) and (4.29) reduce to

$$\begin{aligned}
 \mathbf{A}|_{\mathbf{x}(k) \perp \mathbf{x}(k-1)} &= \frac{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)\mathbf{x}^T(k) - \|\mathbf{x}(k-1)\|^2\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - (\mathbf{x}^T(k)\mathbf{x}(k-1))^2} \\
 &= \frac{-\|\mathbf{x}(k-1)\|^2\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2} \\
 &= \frac{-\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2}
 \end{aligned} \tag{4.55}$$

and

$$\begin{aligned}
 \mathbf{b}|_{\mathbf{x}(k) \perp \mathbf{x}(k-1)} &= n(k) \frac{\|\mathbf{x}(k-1)\|^2\mathbf{x}(k) - (\mathbf{x}^T(k-1)\mathbf{x}(k))\mathbf{x}(k-1)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - (\mathbf{x}^T(k)\mathbf{x}(k-1))^2} \\
 &= n(k) \frac{\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2}
 \end{aligned} \tag{4.56}$$

The coefficient error vector now reduces to

$$\Delta \mathbf{w}(k+1) = \left[\mathbf{I} - P_e(k) \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2} \right] \Delta \mathbf{w}(k) + P_e(k) \frac{n(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \tag{4.57}$$

which is the same as in (4.47) for the case of the SM-NLMS algorithm. Consequently we get

$$\Delta \xi(k+1)|_{\mathbf{x}(k) \perp \mathbf{x}(k-1)} = \left(1 - \frac{2P_e(k) - P_e^2(k)}{N} \right) \Delta \xi(k) + P_e^2(k) \frac{\sigma_n^2}{N+1-\nu_x} \tag{4.58}$$

Appendix A4.3

In this Appendix we show the relation $\delta = \frac{\sigma_x^2}{b} E[\|\Delta \mathbf{w}_\infty\|^2] - \sigma_x^2 E[\Delta \mathbf{w}_\infty^T \mathbf{R} \Delta \mathbf{w}_\infty] \geq 0$ holds true for the autocorrelation matrix in Equation (4.43). Using Equation (4.43) we get

$$\begin{aligned}
\delta &= \frac{\sigma_x^2}{b} E[\|\Delta \mathbf{w}_\infty\|^2] - \sigma_x^2 \Delta \mathbf{w}_\infty^T \begin{bmatrix} 1 & r & r^2 & \dots & r^{N-1} \\ r & 1 & r & \dots & r^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r^{N-1} & r^{N-2} & r^{N-3} & \dots & 1 \end{bmatrix} \Delta \mathbf{w}_\infty \\
&= \frac{\sigma_x^2}{b} E[\|\Delta \mathbf{w}_\infty\|^2] - \sigma_x^2 E[\|\Delta \mathbf{w}_\infty\|^2] - \sigma_x^2 \sum_{j=1}^N \sum_{\substack{i=1 \\ i \neq j}}^N E[\Delta w_i \Delta w_j] r^{|i-j|} \\
&= \frac{\sigma_x^2}{b} \left\{ (1-b) E[\|\Delta \mathbf{w}_\infty\|^2] - b \sum_{j=1}^N \sum_{\substack{i=1 \\ i \neq j}}^N E[\Delta w_i \Delta w_j] r^{|i-j|} \right\} \\
&= \frac{\sigma_x^2}{1-r} \left\{ 2r E[\|\Delta \mathbf{w}_\infty\|^2] - (1-r) \sum_{j=1}^N \sum_{\substack{i=1 \\ i \neq j}}^N E[\Delta w_i \Delta w_j] r^{|i-j|} \right\} \tag{4.59}
\end{aligned}$$

where $b = (1-r)/(1+r)$ was used. Considering the worst-case scenario, we substitute $E[\Delta w_i \Delta w_j]$ with the magnitude of the maximum value of the cross-terms, i.e., $c_{max} = \max_{\forall i,j,i \neq j} |E[\Delta w_i \Delta w_j]|$

$$\begin{aligned}
\delta &\geq \frac{\sigma_x^2}{1-r} \left\{ 2r E[\|\Delta \mathbf{w}_\infty\|^2] - (1-r) \sum_{j=1}^N \sum_{\substack{i=1 \\ i \neq j}}^N c_{max} r^{|i-j|} \right\} \\
&= \frac{\sigma_x^2}{1-r} \left\{ 2r E[\|\Delta \mathbf{w}_\infty\|^2] - (1-r) c_{max} \sum_{i=1}^{N-1} 2(N-i) r^i \right\} \\
&\geq \frac{\sigma_x^2}{1+r} \left\{ 2r E[\|\Delta \mathbf{w}_\infty\|^2] - 2(N-1)(1-r) c_{max} \sum_{i=1}^{N-1} r^i \right\} \\
&= \frac{\sigma_x^2}{1-r} \left\{ 2r E[\|\Delta \mathbf{w}_\infty\|^2] - 2r(N-1) c_{max} (1-r^{N-1}) \right\} \\
&\geq 0 \tag{4.60}
\end{aligned}$$

where $E[\|\Delta \mathbf{w}_\infty\|^2] \geq (N-1)c_{max}$ was used, which holds true for any covariance matrix.

Chapter 5

Set-Membership Affine-Projection

Algorithm

This chapter presents a new data selective adaptive filtering algorithm, the set-membership affine-projection (SM-AP) algorithm. The algorithm generalizes the ideas of the recently proposed set-membership NLMS (SM-NLMS) algorithm and the set-membership BNDRLMS algorithm proposed in the previous chapter to include constraint sets constructed from the past input and desired signal pairs. The resulting algorithm can be seen as a set-membership version of the affine-projection (AP) algorithm with an optimized step size. Also the SM-AP algorithm does not trade convergence speed with misadjustment and computational complexity as most adaptive filtering algorithms. Simulations show the good performance of the algorithm, especially for colored input signals, in terms of convergence, final misadjustment, and reduced computational complexity.

5.1 Introduction

For highly correlated input signals the RLS algorithms are known to present faster convergence than the LMS algorithm and its normalized version, the NLMS algorithm [1]. This advantage comes at the expense of a higher computational complexity. Data-reusing algorithms are known to be a viable alternative to the RLS algorithm in terms of lower computational complexity in situations where the input signal is correlated. The affine-projection

(AP) algorithm [42, 54, 111, 58] is among the prominent unconstrained adaptation algorithms that allow tradeoff between fast convergence and low computational complexity. By adjusting the number of projections, or alternatively, the number of reuses, we can obtain ramping performances from that of the normalized least-mean square (NLMS) algorithm to that of the sliding-window recursive least squares (RLS) algorithm [58, 56]. The penalty to be paid when increasing the number of data reuse is a slight increase in algorithm misadjustment. Trade-off between final misadjustment and convergence speed is achieved through the introduction of a step size, which is not the best solution. An alternative solution to this drawback is to employ the concept of set-membership filtering (SMF) [68, 69, 70, 98, 99] to data reusing algorithms. SMF specifies an upper bound on the estimation error and reduces computational complexity on the average due to its data-discerning property. The set-membership NLMS (SM-NLMS) algorithm proposed in [69] was shown to achieve both fast convergence and low misadjustment, and its data-selectivity and low computational complexity per update makes it very attractive in various applications [70, 108]. An early attempt in this direction was the introduction of the set-membership binormalized data-reusing LMS algorithm (SM-BNDRLMS) in Chapter 4 [77, 78]. This chapter generalizes the ideas in Chapter 4 by adopting P past data-pairs. The resulting algorithms include the SM-NLMS and SM-BNDRLMS as special cases, which correspond to choosing $P = 1$ and $P = 2$, respectively. The conventional affine-projection (AP) algorithm [13, 42, 54] is also shown to be a limiting case of the new algorithms, when the predefined bound of the estimation error goes to zero.

The chapter is organized as follows. The new algorithm, the set-membership affine-projection (SM-AP) algorithm, is derived in Section 5.2. Section 5.3 contains the simulations, and Section 5.4 the concluding remarks.

5.2 Set-Membership Affine-Projection Algorithm

The set-membership binormalized data-reusing (SM-BNDRLMS) algorithms derived in Chapter 4 made use of two members of the exact membership set $\psi(k)$ in (4.4) to con-

struct a set of feasible solutions for the coefficient update. This section introduces the set-membership affine-projection (SM-AP) algorithm whose updates belong to a member of the exact membership set which is spanned by P constraint sets. The necessary concepts and notations of set-membership filtering (SMF) are defined in Chapter 4 of this thesis. For a more complete treatment of SMF, see [70].

5.2.1 Derivation of the General SM-AP Algorithm

Let us start by expressing the exact membership set $\psi(k)$ defined in (4.4) as

$$\psi(k) = \bigcap_{i=1}^{k-P} \mathcal{H}(i) \bigcap_{j=k-P+1}^k \mathcal{H}(j) = \psi^{k-P}(k) \bigcap \psi^P(k) \quad (5.1)$$

where $\psi^P(k)$ is the intersection of the P last constraint sets, and $\psi^{k-P}(k)$ is the intersection of the $k - P$ first constraint sets. The objective is to derive an algorithm whose coefficient update belongs to the last P constraint sets, i.e., $\mathbf{w} \in \psi^P(k)$.

Let $\mathcal{S}_i(k)$ denote the hyperplane which contains all vectors \mathbf{w} such that $d(k - i + 1) - \mathbf{w}^T \mathbf{x}(k - i + 1) = g_i(k)$ for $i = 1, \dots, P$. The next section discusses the choice of the parameters $g_i(k)$ but for the time being all choices satisfying the bound constraint are valid. That is, if all $g_i(k)$ are chosen such that $|g_i(k)| \leq \gamma$ then $\mathcal{S}_i(k) \in \mathcal{H}(k - i + 1)$.

Let us state the following optimization criterion for the vector update whenever $\mathbf{w}(k) \notin \psi^P(k)$

$$\begin{aligned} \mathbf{w}(k+1) &= \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to:} \\ \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w} &= \mathbf{g}(k) \end{aligned} \quad (5.2)$$

where $\mathbf{d}(k) \in \mathbb{R}^{P \times 1}$ contains the desired outputs from the P last time instants, $\mathbf{g}(k) \in \mathbb{R}^{P \times 1}$ specifies the point in $\psi^P(k)$, and $\mathbf{X}(k) \in \mathbb{R}^{N \times P}$ contains the corresponding input vectors, i.e.,

$$\begin{aligned} \mathbf{g}(k) &= [g_1(k) \ g_1(k) \ \dots \ g_P(k)]^T \\ \mathbf{d}(k) &= [d(k) \ d(k-1) \ \dots \ d(k-P+1)]^T \\ \mathbf{X}(k) &= [\mathbf{x}(k) \ \mathbf{x}(k-1) \ \dots \ \mathbf{x}(k-P+1)] \end{aligned} \quad (5.3)$$

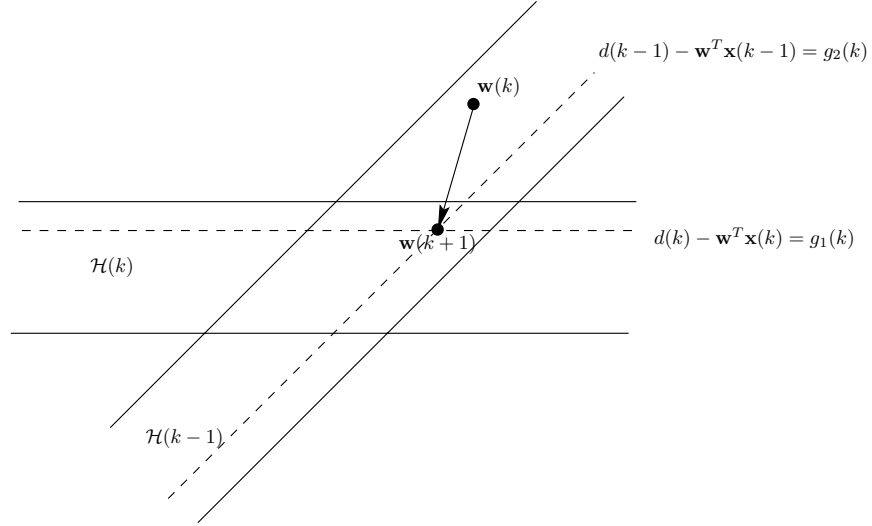


Figure 5.1: General algorithm update.

where $\mathbf{x}(k)$ is the input-signal vector

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \dots \ x(k-N+1)]^T. \quad (5.4)$$

Figure 5.1 shows the update procedure for a two-dimensional problem. Using the method of Lagrange multipliers, the unconstrained function to be minimized is

$$J_{\mathbf{w}} = \|\mathbf{w} - \mathbf{w}(k)\|^2 + \boldsymbol{\lambda}^T(k) [\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w} - \mathbf{g}(k)] \quad (5.5)$$

where $\boldsymbol{\lambda}(k) \in \mathbb{R}^{P \times 1}$ is a vector of Lagrange multipliers. After setting the gradient of $J_{\mathbf{w}}$ with respect to \mathbf{w} equal to zero, we get

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k)\boldsymbol{\lambda}(k) \quad (5.6)$$

Invoking the constraints in (5.2), we obtain

$$\begin{aligned} \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k+1) &= \mathbf{g}(k) \\ &= \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k) - \mathbf{X}^T(k)\mathbf{X}(k)\boldsymbol{\lambda}(k) \end{aligned} \quad (5.7)$$

and consequently we get,

$$\mathbf{X}^T(k)\mathbf{X}(k)\boldsymbol{\lambda}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k) - \mathbf{g}(k) = \mathbf{e}(k) - \mathbf{g}(k) \quad (5.8)$$

where

$$\mathbf{e}(k) = [e(k) \ \epsilon(k-1) \ \dots \ \epsilon(k-P+1)]^T \quad (5.9)$$

with $\epsilon(k-i) = d(k-i) - \mathbf{x}^T(k-i)\mathbf{w}(k)$ denoting the *a posteriori* error at iteration $k-i$. The update equation is now given by (5.6) with $\boldsymbol{\lambda}(k)$ being the solution to the system of equations given in (5.8), i.e.,

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}(k) [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} [\mathbf{e}(k) - \mathbf{g}(k)] & \text{if } |e(k)| > \gamma \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \quad (5.10)$$

Remark 1. To evaluate if an update $\mathbf{w}(k+1)$ is required, it is only necessary to check if $\mathbf{w}(k) \notin \mathcal{H}(k)$. This is a consequence of the constraint set reuse guaranteeing that $\mathbf{w}(k) \in \mathcal{H}(k-i+1)$ holds for $i = 2, \dots, P$ before an update.

Remark 2. During initialization, i.e., for time instants $k < P$ only knowledge of $\mathcal{H}(i)$ for $i = 1, \dots, k$ can be assumed. If an update is needed for the initial time instants $k < P$, the algorithm is used with the k available constraint sets.

Remark 3. We can easily verify that choosing the bound $\gamma = 0$, the algorithm will reduce to the conventional AP algorithm [42] with unity step size.

5.2.2 The Parameter Vector $\mathbf{g}(k)$

So far the only requirement on the parameters $g_i(k)$ has been that they should be points in $\mathcal{H}(k-i+1)$, i.e., $|g_i(k)| \leq \gamma$. Obviously there is an infinite number of possible choices for $g_i(k)$, each one leading to a different update.

Choice 1:

A trivial choice would be $\mathbf{g}(k) = \mathbf{0}$, i.e., to force the *a posteriori* errors to be zero at the last P time instants. Inserting $\mathbf{g}(k) = \mathbf{0}$ in (5.8) and solving for $\boldsymbol{\lambda}(k)$ leads to the recursions

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}(k) [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} \mathbf{e}(k) & \text{if } |e(k)| > \gamma \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \quad (5.11)$$

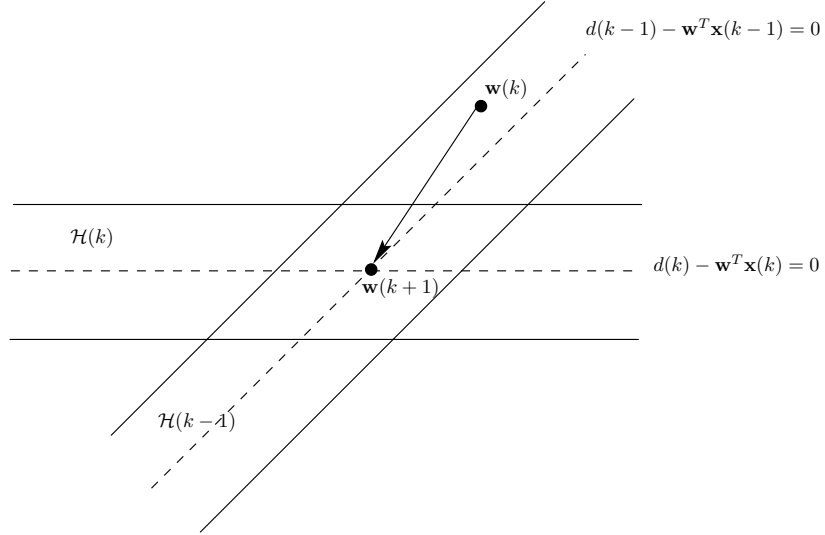


Figure 5.2: Update resulting in zero a posteriori error.

The updating equation (5.11) is identical to the conventional affine-projection (AP) algorithm with unity step size whenever $\mathbf{w}(k) \notin \mathcal{H}(k)$. The approach taken here allows a considerable reduction in average complexity as compared with the conventional AP algorithm due to the data selectivity. Figure 5.2 shows a graphical view of the coefficient update.

Choice 2:

We now take a closer look at Equation (5.8) that solves for $\boldsymbol{\lambda}(k)$ and make some important observations. We know already that $\mathbf{w}(k) \in \mathcal{H}(k-i+1)$, i.e., $|\epsilon(k-i+1)| = |d(k-i+1) - \mathbf{x}^T(k-i+1)\mathbf{w}(k)| \leq \gamma$, for $i = 2, \dots, P$. Therefore, choosing $g_i(k) = \epsilon(k-i+1)$, for $i \neq 1$, will cancel all but the first element on the right-hand side of (5.8). Now we only need to choose the constraint value $g_1(k)$. In the same way as the SM-NLMS we can choose $g_1(k)$ such that the solution lies at the nearest boundary of $\mathcal{H}(k)$, i.e., $g_1(k) = \gamma e(k)/|e(k)|$. With these choices, Equation (5.8) reduces to

$$\mathbf{X}^T(k)\mathbf{X}(k)\boldsymbol{\lambda}(k) = \alpha(k)e(k)\mathbf{u}_1 \quad (5.12)$$

where $\alpha(k) = 1 - \frac{\gamma}{|e(k)|}$ and $\mathbf{u}_1 = [1 \ 0 \ \dots \ 0]^T$. Finally we can write the update equation as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k) [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} \alpha(k)e(k)\mathbf{u}_1 \quad (5.13)$$

$$\alpha(k) = \begin{cases} 1 - \frac{\gamma}{|e(k)|}, & \text{if } |e(k)| > \gamma \\ 0, & \text{otherwise} \end{cases} \quad (5.14)$$

The last algorithm minimizes the Euclidean distance $\|\mathbf{w} - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w} \in \psi^P(k)$ such that the *a posteriori* error at iteration $k - i$, $\epsilon(k - i) = d(k - i) - \mathbf{w}^T(k)\mathbf{x}(k - i)$, is kept constant for $i = 2, \dots, P$. The updating procedure is shown graphically in Figure 5.3. The equations of the SM-AP algorithm are summarized in Table 5.1.

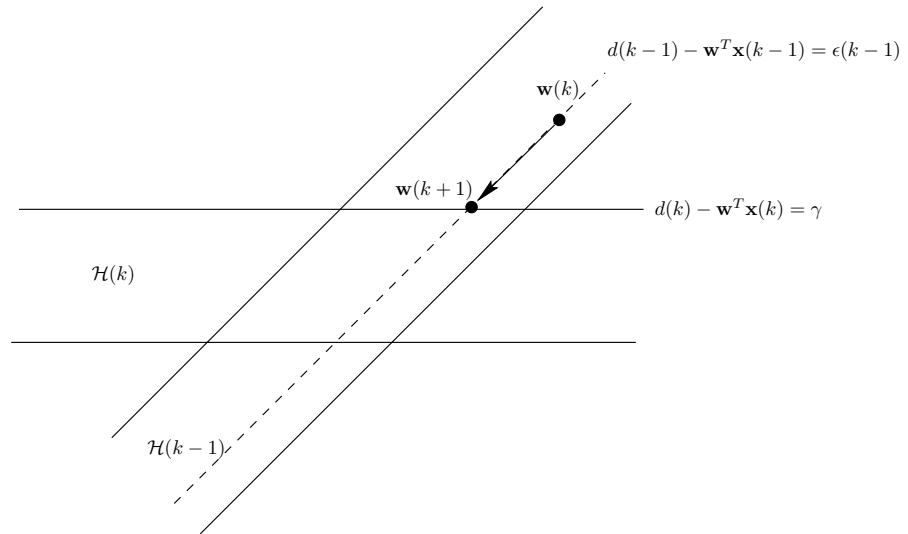


Figure 5.3: Update resulting in constant *a posteriori* error.

5.3 Simulation Results

5.3.1 System Identification

The SM-AP algorithm, which updates along constant *a posteriori* errors (Choice 2) was used to identify a system of order $N = 10$. The input signal was colored noise, generated by filtering Gaussian noise through the fourth-order IIR filter $x(k) = 0.95x(k-1) + 0.19x(k-2) + 0.09x(k-3) - 0.5x(k-4)$ [112]. The signal-to-noise ratio (*SNR*) was set to 80dB, and the bound was chosen to $\gamma = \sqrt{5\sigma_n^2}$ where σ_n^2 is the variance of the additional noise. Figure 5.4 shows the learning curves averaged over

Table 5.1: The Set-Membership Affine-Projection Algorithm.

| SM-AP Algorithm |
|---|
| <pre> for each k { $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { $\alpha(k) = 1 - \gamma/ e(k)$ $\mathbf{t}(k) = [\mathbf{X}^T(k)\mathbf{X}(k)]^{-1} \alpha(k)e(k)\mathbf{u}_1$ $\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k)\mathbf{t}(k)$ } else { $\mathbf{w}(k+1) = \mathbf{w}(k)$ } } </pre> |

500 simulations for $P = 1, 2, 4$, and 6 constraint set reuses. The number of data reuses in the conventional AP algorithm was chosen as $P = 4$.

Figure 5.4 clearly shows the increase in convergence speed obtained by increasing P . As can be seen from the figure all curves of the SM-AP algorithms have the same level of misadjustment, which is lower than that of the the conventional AP algorithm. In an ensemble of 500 experiments and 1000 iterations, the average number of updates per experiment for the SM-AP algorithm were 962, 402, 268, and 215 out of 1000 for $P = 1$, $P = 2$, $P = 4$, and $P = 6$ respectively, demonstrating the significant reduction of computational complexity obtained by increasing P .

5.3.2 Adaptive Equalizer

The SM-AP algorithm (Choice 2) was used to adapt an equalizer in a single-user system using BPSK transmission. The channel was a three-tap raised-cosine channel [12] having the impulse response $\{0.3887, 1, 0.3887\}$. The number of taps in the equalizer was chosen to $N = 11$, and the delay associated with the reference signal was chosen as $D = 7$ (see Chapter 1 for a schematic diagram of the equalizer setup). The threshold γ was chosen to $\sqrt{9\sigma_n^2}$ according to the choice made in [98].

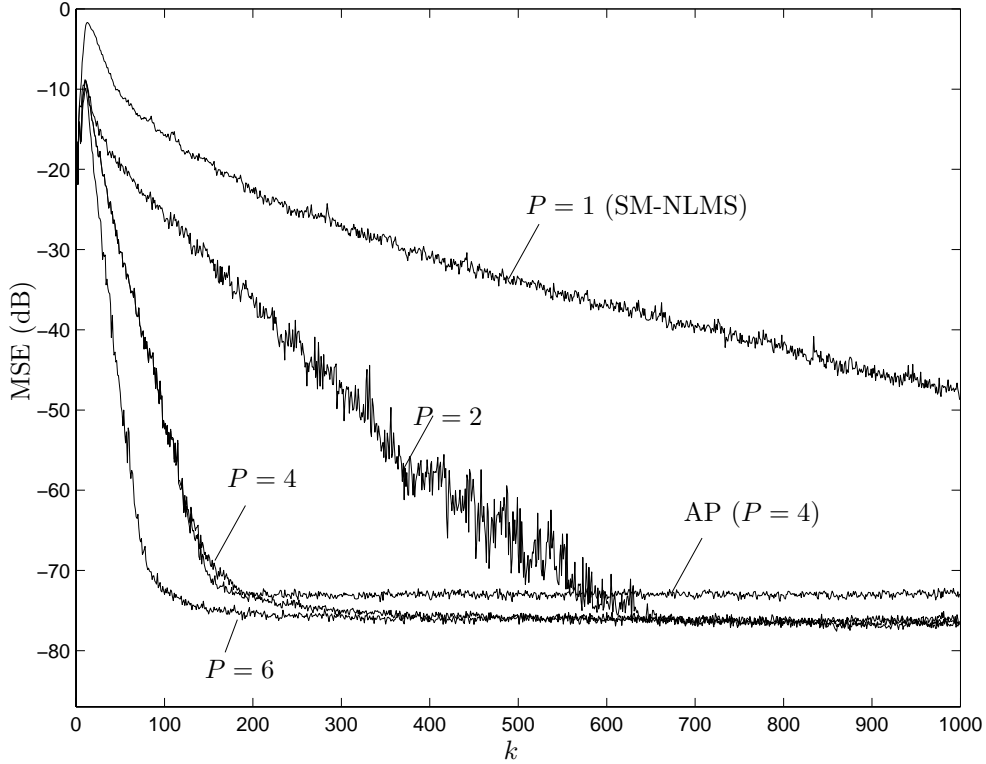


Figure 5.4: Learning curves for the SM-AP algorithm with $P = 1$, $P = 2$, $P = 4$, $P = 6$ and the conventional AP algorithm with $P = 4$, $\gamma = \sqrt{5\sigma_n^2}$, $SNR = 80$ dB, and colored input signal.

Figure 5.5 shows the learning curves for the SM-AP algorithm using different values of P , the SM-BNDRLMS-I algorithm proposed in Chapter 5, and the conventional NLMS algorithm. The SNR was set to 30 dB. In an ensemble of 500 experiments and 1000 iterations, the average number of updates per experiment for the SM-AP algorithm were 390, 320, and 340 for $P = 1$, $P = 2$, $P = 3$. The average number of updates for SM-BNDRLMS-I algorithm was 310, where 190 were full updates, i.e., 190 out of 310 updates required two steps to be performed. We can see from the figure that the largest performance improvement is obtained when going from $P = 1$ to $P = 2$ data reuses. A small penalty in terms of an increased MSE is observed for $P = 3$.

Figure 5.6 shows the learning curves for the case when the SNR was set to 15 dB. The figure indicates that the performance improvement decreases for a decreasing SNR . A small improvement in the convergence speed can be observed for $P = 2$ as compared to the case

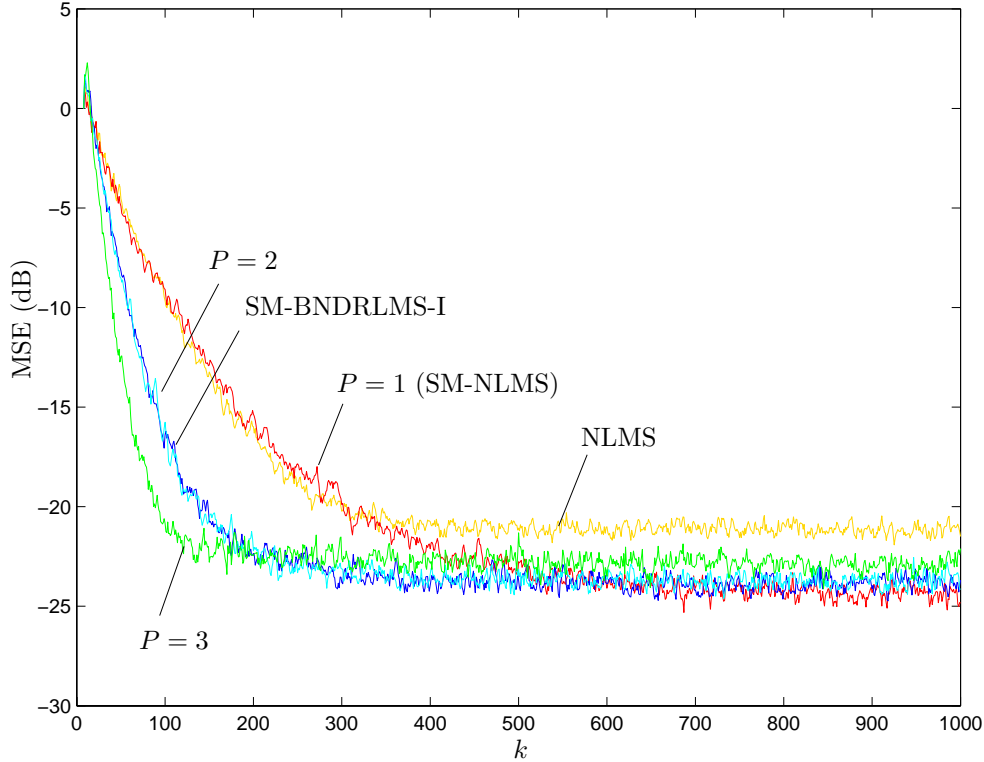


Figure 5.5: Learning curves for the adaptive equalizer using the SM-AP algorithm with $P = 1$, $P = 2$, $P = 3$, the SM-BNDRLMS-I, and the conventional NLMS algorithm with $\mu = 1$, $\gamma = \sqrt{9\sigma_n^2}$, $SNR = 30$ dB.

of $P = 1$ (SM-NLMS). For this SNR, no performance improvement in terms of convergence speed can be observed for $P > 2$. The average number of updates per experiment for the SM-AP algorithm were 230, 210, and 225 for $P = 1$, $P = 2$, and $P = 3$, respectively. The average number of updates for SM-BNDRLMS-I algorithm was 210, where 115 were full updates, i.e., 115 out of 210 updates required two steps to be performed.

5.4 Conclusions

A novel data-selective adaptation algorithm, the set-membership affine-projection (SM-AP) algorithm was derived based on the concept of set-membership filtering. The algorithm utilizes consecutive data-pairs in order to construct a space of feasible solutions for the updates. The SM-AP algorithm reduces the tradeoff of misadjustment and computational

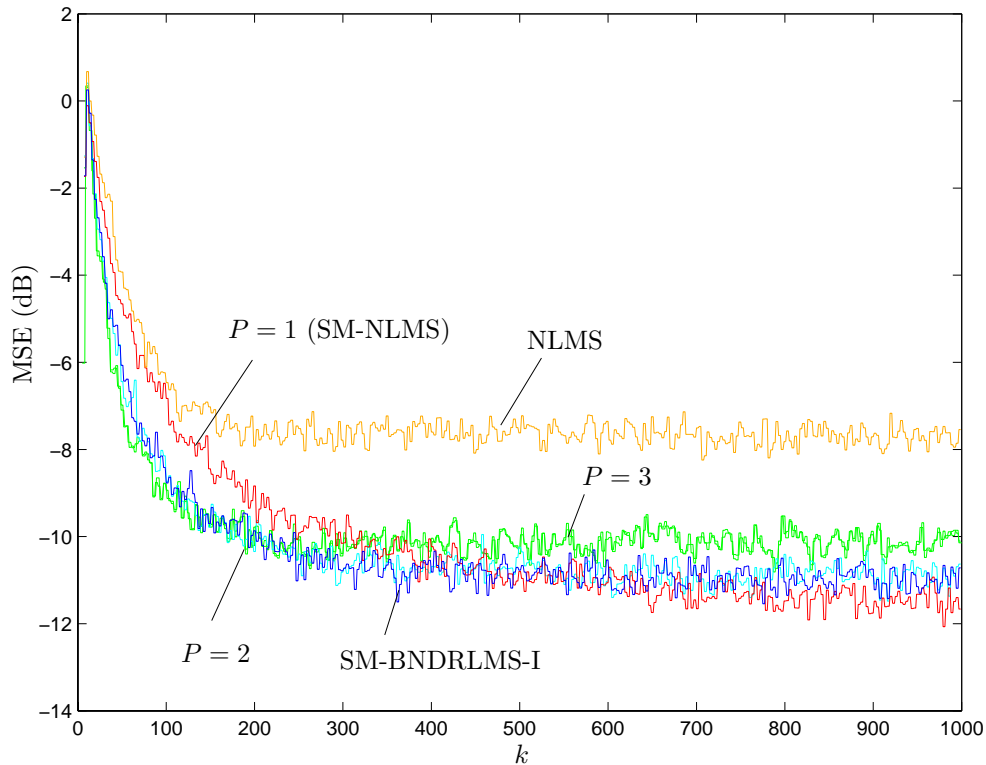


Figure 5.6: Learning curves for the adaptive equalizer using the SM-AP algorithm with $P = 1$, $P = 2$, $P = 3$, the SM-BNDRLMS-I, and the conventional NLMS algorithm with $\mu = 1$, $\gamma = \sqrt{9\sigma_n^2}$, $SNR = 15$ dB.

complexity associated with the conventional AP algorithm. Simulations confirmed that the proposed algorithm leads to fast convergence speed, low misadjustment, and a substantial reduction in the number of updates. The more correlated is the input signal, the better is the performance of the SM-AP algorithm when compared with the SM-NLMS algorithm. This improvement is more clearly observed in cases of high SNR.

Chapter 6

Low-Complexity Constrained Affine-Projection Algorithms

This chapter presents low-complexity constrained affine-projection algorithms. An efficient implementation of the constrained affine-projection algorithm utilizing Householder transformation is presented. A data-selective version of the constrained algorithm is also derived based on the concept of set-membership filtering. The data-selective property can greatly reduce the average computational burden as compared with a nonselective approach without compromising speed of convergence and final misadjustment. The chapter also discusses important aspects of convergence and stability of constrained normalized adaptation algorithms in general. Computer simulations are also included providing extra insight to the algorithm behavior.

6.1 Introduction

In Chapters 2 and 3, we derived and discussed adaptive filtering algorithms satisfying linear constraints. This class of algorithms was mentioned to have various applications in fields such as communications and system identification. The algorithms considered were mainly LMS-type or RLS-type algorithms. As already discussed in Chapter 2, the constrained LMS (CLMS) algorithm is attractive due to the low computational complexity. The main drawback of the CLMS algorithm is the slow convergence speed for colored input

signals [27]. On the other hand, the constrained RLS (CRLS) enjoys a fast convergence but at the expense of high computational complexity. Similarly to the case of unconstrained adaptation algorithms, it would be of interest to have linearly-constrained adaptive filtering algorithms which could, by using a suitable number of data-reuses, balance convergence speed and computational complexity.

In this chapter we develop and analyze a constrained affine-projection (CAP) algorithm using the same framework already used for other normalized constrained algorithms, such as the normalized constrained LMS (NCLMS) and binormalized data-reusing constrained LMS (BNDRCLMS) algorithms [86]. Moreover, a Householder transformation introduced in [66] is used to derive an efficient implementation for the CAP algorithm. Thereafter, the ideas of normalized constrained algorithms are extended to the framework of set-membership filtering (SMF) [69], from which a set-membership constrained affine-projection (SM-CAP) algorithm is derived. The SM-CAP algorithm, which can also be seen as a constrained version of the set-membership affine-projection (SM-AP) algorithm [113], retains the fast convergence of the CAP algorithm, and low misadjustment is obtained due to the data-selective property. The *a posteriori* output constrained LMS (APOC-LMS) algorithm proposed in [71] bears similarity to the proposed SM-CAP algorithm for the special case of one data-reuse and a single constraint. However, our approach differs from that in [71] by the use of a correction term that prevents accumulation of errors when implemented in finite precision. We further analyze the bias of the coefficient-error vector of the proposed algorithms.

The chapter is organized as follows. Section 6.2 presents the derivation of the CAP algorithm, followed by an efficient Householder implementation in Section 6.3. Section 6.4 presents the SM-CAP algorithm. Section 6.5 analyze the bias of the coefficient-error vector for the proposed algorithms. Simulations of the algorithms are shown in Section 6.6, followed by conclusions in Section 6.7.

6.2 The Constrained Affine-Projection Algorithm

As was discussed in Chapter 2, the constraints in linearly constrained adaptive filtering are given by the following set of p equations

$$\mathbf{C}^T \mathbf{w} = \mathbf{f} \quad (6.1)$$

where \mathbf{C} is an $N \times p$ constraint matrix (parameter M used in Chapter 2 assumed here to be unity) and \mathbf{f} is a vector containing the p constraint values.

The constrained affine-projection (CAP) algorithm to be derived solves the following optimization problem

$$\begin{aligned} \mathbf{w}(k+1) &= \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to} \\ \mathbf{C}^T \mathbf{w} &= \mathbf{f} \\ \mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w} &= \mathbf{0} \end{aligned} \quad (6.2)$$

where $\mathbf{d}(k) \in \mathbb{R}^{P \times 1}$ and $\mathbf{X}(k) \in \mathbb{R}^{N \times P}$ are the desired-signal vector and input-signal matrix defined by (1.7). Using the Lagrange multipliers in the following objective function

$$J_{\mathbf{w}} = \|\mathbf{w} - \mathbf{w}(k)\|^2 + \boldsymbol{\lambda}_1^T [\mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w}] + \boldsymbol{\lambda}_2^T [\mathbf{C}^T \mathbf{w} - \mathbf{f}] \quad (6.3)$$

the CAP algorithm becomes:

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) + \mathbf{X}(k) \mathbf{t}(k)] + \mathbf{F} \quad (6.4)$$

with

$$\mathbf{t}(k) = [\mathbf{X}^T(k) \mathbf{P} \mathbf{X}(k)]^{-1} \mathbf{e}(k) \quad (6.5)$$

and

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w}(k) \quad (6.6)$$

As in Chapter 2, the matrix

$$\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \quad (6.7)$$

Table 6.1: The Constrained Affine-Projection Algorithm.

| CAP Algorithm |
|--|
| for each k { $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k)$ $\mathbf{t}(k) = [\mathbf{X}^T(k) \mathbf{P} \mathbf{X}(k) + \delta \mathbf{I}]^{-1} \mathbf{e}(k)$ $\mathbf{w}(k+1) = \mathbf{P} [\mathbf{w}(k) + \mu \mathbf{X}(k) \mathbf{t}(k)] + \mathbf{F}$ } |

is a projection matrix for a projection onto the homogeneous hyperplane defined by $\mathbf{C}^T \mathbf{w}(k) = \mathbf{0}$, and the vector

$$\mathbf{F} = \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} \mathbf{f} \quad (6.8)$$

is used to move the projected solution back to the constraint hyperplane.

For $P = 1$ or $P = 2$, the above relations will result in the normalized constrained LMS (NCLMS) or binormalized data-reusing constrained LMS (BNDRCLMS) algorithms [86], respectively. For all constrained algorithms mentioned here, the simplification $\mathbf{P}\mathbf{w}(k) + \mathbf{F} = \mathbf{w}(k)$ should be avoided in a finite precision environment, since accumulation of round-off errors may cause the solution to drift away from the constraint hyperplane [27]. The equations of the constrained affine-projection algorithm are summarized in Table 6.1.

6.3 The Householder-Transform CAP Algorithm

In this section, the results of the Householder-transform constrained LMS (HCLMS) algorithm [66] proposed in Chapter 3 are used to obtain an efficient Householder-transform constrained AP (HCAP) algorithm.

From Chapter 3 we remember that an orthogonal matrix \mathbf{Q} was used to transform the adaptive-filter coefficient vector in order to generate a modified coefficient vector $\bar{\mathbf{w}}(k) = \mathbf{Q}\mathbf{w}(k)$. If the same transformation is applied to the input signal $\mathbf{x}(k)$, i.e., $\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k)$, the output signal from the transformed filter $\bar{y}(k) = \bar{\mathbf{w}}^T(k)\bar{\mathbf{x}}(k)$ will be the same as the

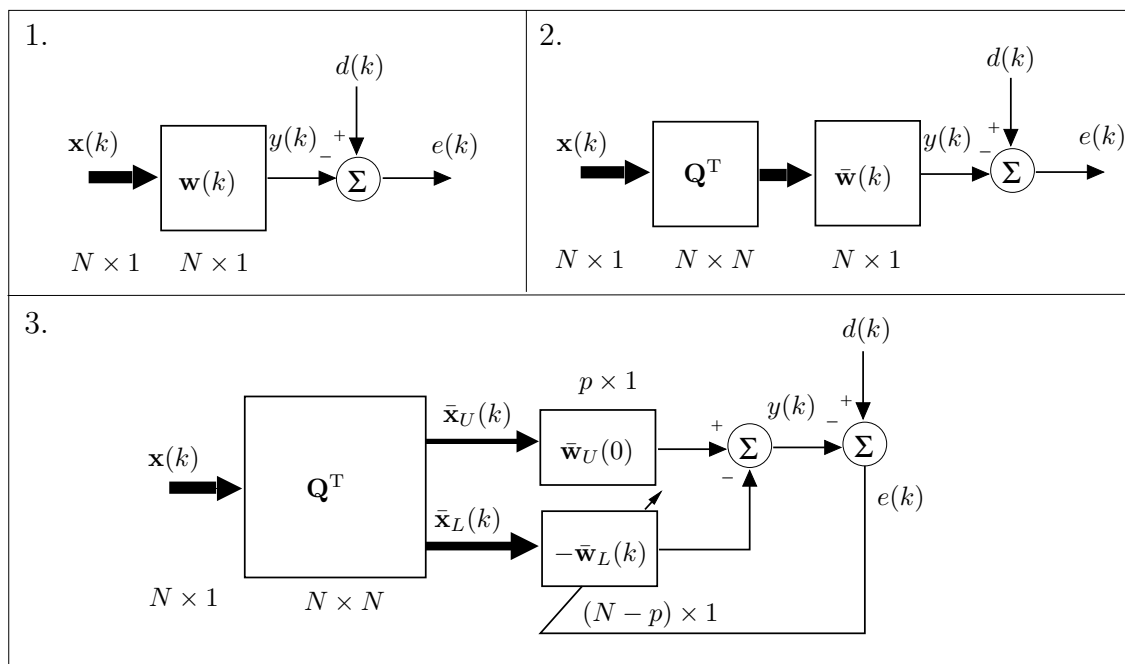


Figure 6.1: The Householder transformation on the coefficient vector.

1. Untransformed adaptive filter
2. Applying the transformation as in (6.9);
3. Splitting the transformed vector $\bar{\mathbf{w}}(k)$ into a constant vector $\bar{\mathbf{w}}_U(0) \in \mathbb{R}^{p \times 1}$ and a vector $\bar{\mathbf{w}}_L(k) \in \mathbb{R}^{(N-p) \times 1}$ to be updated adaptively.

original untransformed filter, i.e., we have

$$\bar{y}(k) = \bar{\mathbf{w}}^T(k)\bar{\mathbf{x}}(k) = \mathbf{w}^T(k)\mathbf{Q}^T\mathbf{Q}\mathbf{x}(k) = \mathbf{w}^T(k)\mathbf{x}(k)$$

If the matrix \mathbf{Q} is chosen such that it triangularize $\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-\frac{1}{2}}$ through a sequence of Householder transformations, as was proposed in Chapter 3 and [66], the first p elements $\bar{\mathbf{w}}_U(0)$ of the transformed vector are constant, while its last $N - p$ elements $\bar{\mathbf{w}}_L(k)$ can be updated using any desired adaptive algorithm. The transformation steps are shown in Fig. 6.1. The advantage of such approach, outlined above, as compared to the generalized sidelobe canceling (GSC) structure was discussed in Chapter 3 and lies in the efficient implementation of the product $\mathbf{Q}\mathbf{x}(k)$ that can be carried out through the following product of p matrices

$$\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k) = \mathbf{Q}_p \cdots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{x}(k) \quad (6.9)$$

Table 6.2: The Householder-Transform Constrained Affine-Projection Algorithm

| HCAP Algorithm |
|--|
| Initialize: $\bar{\mathbf{w}}(0) = \mathbf{Q}\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$ $\bar{\mathbf{w}}_U(0) = p$ first elements of $\bar{\mathbf{w}}(0)$ $\bar{\mathbf{w}}_L(0) = N - p$ last elements of $\bar{\mathbf{w}}(0)$ for each k { $\bar{\mathbf{x}}(k) = \mathbf{Q}\mathbf{x}(k)$ % according to, e.g., [94] $\bar{\mathbf{x}}_U(k) = p$ first elements of $\bar{\mathbf{x}}(k)$ $\bar{\mathbf{x}}_L(k) = N - p$ last elements of $\bar{\mathbf{x}}(k)$ $\bar{\mathbf{w}}(k) = \begin{bmatrix} \bar{\mathbf{w}}_U(0) \\ \bar{\mathbf{w}}_L(k) \end{bmatrix}$ $\bar{\mathbf{d}}(k) = \begin{bmatrix} \bar{\mathbf{w}}_U^T(0)\bar{\mathbf{x}}_U(k) \\ \text{first } P - 1 \text{ elements of } \bar{\mathbf{d}}(k - 1) \end{bmatrix}$ $\mathbf{e}(k) = \mathbf{d}(k) - [\bar{\mathbf{d}}(k) + \bar{\mathbf{X}}_L^T(k)\bar{\mathbf{w}}_L(k)]$ $\bar{\mathbf{t}}(k) = [\bar{\mathbf{X}}_L^T(k)\bar{\mathbf{X}}_L(k) + \delta\mathbf{I}]^{-1}\mathbf{e}(k)$ $\bar{\mathbf{w}}_L(k + 1) = \bar{\mathbf{w}}_L(k) + \mu\bar{\mathbf{X}}_L(k)\bar{\mathbf{t}}(k)$ } |

where

$$\mathbf{Q}_i = \begin{bmatrix} \mathbf{I}_{i-1 \times i-1} & \mathbf{0}^T \\ \mathbf{0} & \bar{\mathbf{Q}}_i \end{bmatrix} \quad (6.10)$$

and matrix $\bar{\mathbf{Q}}_i = \mathbf{I} - 2\bar{\mathbf{v}}_i\bar{\mathbf{v}}_i^T$ is an ordinary $(N - i + 1) \times (N - i + 1)$ Householder transformation matrix. For the implementation of the product in (6.9), see Tables 3.2 and 3.3 in Chapter 3. Table 6.2 presents the Householder-transform constrained affine-projection (HCAP) algorithm.

6.4 Set-Membership Constrained Affine-Projection Algorithm

For a review of the set-membership filtering (SMF) framework and the notation used in this section, see Chapter 4 of this thesis. More detailed treatment of SMF can be found

in [70]. In our SMF formulation we want to design the filter such that the magnitude of estimation error is bounded. For this formulation we express the exact membership set in Equation (4.4) as $\psi(k) = \psi^{k-P}(k) \cap \psi^P(k)$ where $\psi^P(k) = \bigcap_{i=k-P+1}^k \mathcal{H}(i)$, as in the SM-AP algorithm of Chapter 5, which corresponds to the intersection of the P past constraint sets. Next we consider the derivation of a data-selective algorithm whose coefficients belong to the hyperplane defined by equation (6.1) and also to the last P constraint sets, i.e., $\mathbf{C}^T \mathbf{w} = \mathbf{f}$ and $\mathbf{w} \in \psi^P(k)$. Let us state the following optimization criterion whenever $\mathbf{w}(k) \notin \psi^P(k)$.

$$\begin{aligned} \mathbf{w}(k+1) &= \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to} \\ \mathbf{C}^T \mathbf{w} &= \mathbf{f} \\ \mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w} &= \mathbf{g}(k) \end{aligned} \quad (6.11)$$

where $\mathbf{d}(k)$ and $\mathbf{X}(k)$ are given by (1.7) and

$$\mathbf{g}(k) = [g_1(k) \ g_2(k) \ \dots \ g_P(k)]^T \quad (6.12)$$

In order to guarantee that $\mathbf{w}(k+1) \in \psi^P(k)$ the elements of $\mathbf{g}(k)$ are chosen such that $|g_i(k)| \leq \gamma$ for $i = 1 \dots P$. The solution is obtained by applying the method of Lagrange multipliers to the unconstrained function

$$J_{\mathbf{w}} = \|\mathbf{w} - \mathbf{w}(k)\|^2 + \boldsymbol{\lambda}_1^T [\mathbf{f} - \mathbf{C}^T \mathbf{w}] + \boldsymbol{\lambda}_2^T [\mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w} - \mathbf{g}(k)] \quad (6.13)$$

for which the solution is

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{P} [\mathbf{w}(k) + \mathbf{X}(k) \mathbf{t}(k)] + \mathbf{F} \\ &= \mathbf{w}(k) + \mathbf{P} \mathbf{X}(k) \mathbf{t}(k) + \mathbf{C} (\mathbf{C}^T \mathbf{C})^{-1} [\mathbf{f} - \mathbf{C}^T \mathbf{w}(k)] \end{aligned} \quad (6.14)$$

where

$$\begin{aligned} \mathbf{t}(k) &= [\mathbf{X}^T(k) \mathbf{P} \mathbf{X}(k)]^{-1} [\mathbf{d}(k) - \mathbf{X}^T(k) \mathbf{w}(k) - \mathbf{g}(k)] \\ &= [\mathbf{X}^T(k) \mathbf{P} \mathbf{X}(k)]^{-1} [\mathbf{e}(k) - \mathbf{g}(k)] \end{aligned} \quad (6.15)$$

$$\mathbf{e}(k) = [e(k) \ \epsilon(k-1) \ \dots \ \epsilon(k-P+1)]^T \quad (6.16)$$

and $\epsilon(k-i) = d(k-i) - \mathbf{x}(k-i)^T \mathbf{w}(k)$ denoting the *a posteriori* error at iteration $k-i$, with \mathbf{P} and \mathbf{F} given by (6.7) and (6.8) respectively.

Similarly to the SM-AP algorithm discussed in Chapter 5, we have $\mathbf{w}(k) \in \mathcal{H}(k-i+1)$, i.e., $|\epsilon(k-i+1)| \leq \gamma$, for $i \neq 1$. Therefore, choosing $g_i(k) = \epsilon(k-i+1)$, for $i \neq 1$, will cancel all but the first element in the vector $\mathbf{e}(k) - \mathbf{g}(k)$ of (6.15).

The first element of $\mathbf{g}(k)$ is chosen such that the *a posteriori* error lies on the closest boundary of $\mathcal{H}(k)$, i.e., $g_1(k) = \gamma e(k)/|e(k)|$. With the above choices we get

$$\mathbf{t}(k) = [\mathbf{X}^T(k)\mathbf{P}\mathbf{X}(k)]^{-1} \alpha(k)e(k)\mathbf{u}_1 \quad (6.17)$$

where $\mathbf{u}_1 = [1 \ 0 \ \dots \ 0]^T$ and

$$\alpha(k) = \begin{cases} 1 - \gamma/|e(k)| & \text{if } |e(k)| > \gamma \\ 0 & \text{otherwise} \end{cases} \quad (6.18)$$

is the data dependent step size. Note that for time instants $k < P$ only knowledge of $\mathcal{H}(i)$ for $i = 1, \dots, k$ can be assumed. If an update is needed for the initial time instants $k < P$, the algorithm is used with the k available constraint sets. The equations of the SM-CAP algorithm are summarized in Table 6.3 and a graphical description in \mathbb{R}^2 is shown in Figure 6.2 for the case of $N = 2$ filter coefficients and $P = 1$ data-reuse.

We note that for the particular case of $P = 1$, the SM-CAP reduces to

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\alpha(k)e(k)}{\mathbf{x}^T \mathbf{P} \mathbf{x}(k)} \mathbf{P} \mathbf{x}(k) + \mathbf{C}(\mathbf{C}^T \mathbf{C})^{-1} [\mathbf{f} - \mathbf{C}^T \mathbf{w}(k)] \quad (6.19)$$

which for a single constraint ($p = 1$), apart from the correction term, is identical to the *a posteriori* output constrained LMS (APOC-LMS) algorithm proposed in [71], reproduced below:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\alpha(k)e(k)}{\mathbf{x}^T \mathbf{P} \mathbf{x}(k)} \mathbf{P} \mathbf{x}(k) \quad (6.20)$$

We stress that in our formulation no accumulation of roundoff errors will cause the solution to drift away from the constraint hyperplane.

Table 6.3: The Set-Membership Constrained Affine-Projection Algorithm.

| SM-CAP Algorithm |
|--|
| <pre> for each k { $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { $\alpha(k) = 1 - \gamma/ e(k)$ $\mathbf{t}(k) = [\mathbf{X}^T(k) \mathbf{P} \mathbf{X}(k) + \delta \mathbf{I}]^{-1} \alpha(k)e(k)\mathbf{u}_1$ $\mathbf{w}(k+1) = \mathbf{P} [\mathbf{w}(k) + \mathbf{X}(k) \mathbf{t}(k)] + \mathbf{F}$ } else { $\mathbf{w}(k+1) = \mathbf{w}(k)$ } } </pre> |

6.5 On the Convergence of the CAP Algorithms

For unconstrained adaptation algorithms, it is usually expected that convergence of the coefficients in the mean can be assured as the number of iterations goes to infinity. For normalized algorithms, such as the NLMS, BNDRLMS, or quasi-Newton (QN) [60] algorithms, convergence with probability one is usually more tractable and is sometimes preferred. As the CAP and SM-CAP algorithms are normalized algorithms, we will favor the latter approach in the analysis to be presented in this section. We will analyze the bias for a system identification problem where a known training sequence is available. An example of such a setup is given in [36] where the plant is constrained to have linear phase.

The optimal solution \mathbf{w}_{opt} to the constrained optimization problem is given by [89]

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{C} (\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C})^{-1} (\mathbf{C}^T\mathbf{R}^{-1}\mathbf{p} - \mathbf{f}). \quad (6.21)$$

Let $\mathbf{d}(k)$ be modeled as

$$\mathbf{d}(k) = \mathbf{X}^T(k)\mathbf{w}_{opt} \quad (6.22)$$

If the coefficient-error vector is defined as

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_{opt} \quad (6.23)$$

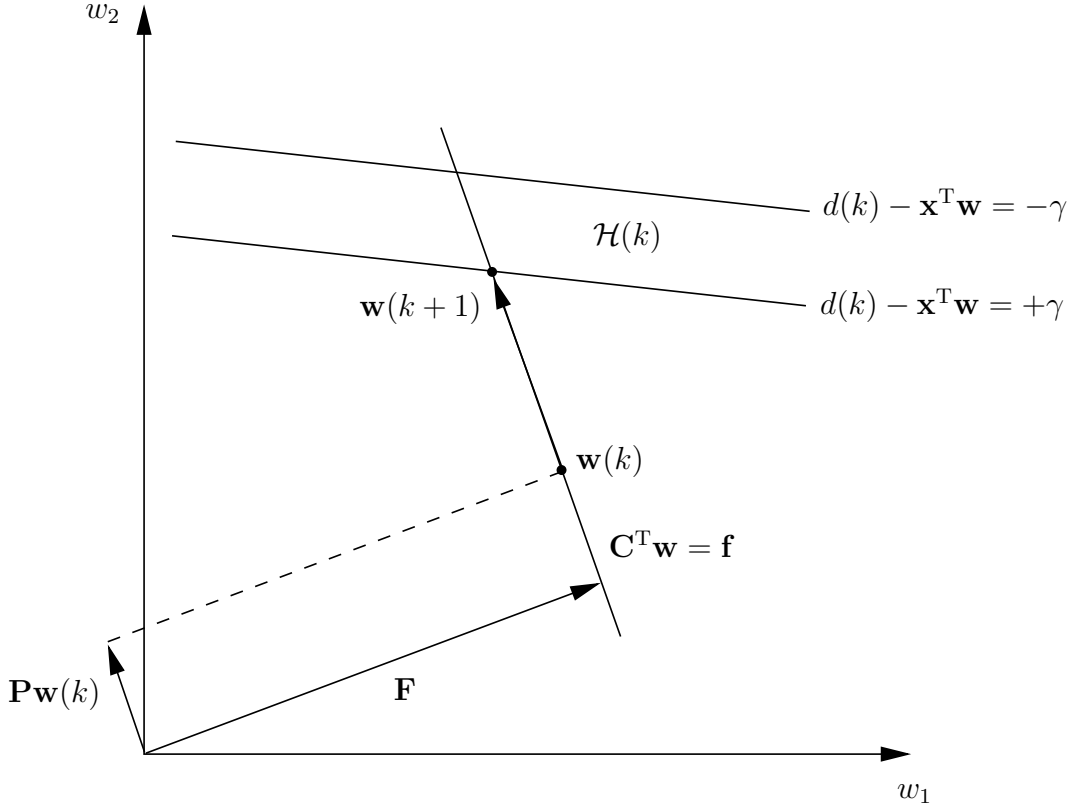


Figure 6.2: Geometrical interpretation of the SM-CAP algorithm in \mathbb{R}^2 for $N = 2$ and $P = 1$.

we get

$$\begin{aligned} \Delta \mathbf{w}(k+1) &= \mathbf{P}\{\mathbf{I} - \mu \mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\}\Delta \mathbf{w}(k) \\ &\quad + \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}[\mathbf{f} - \mathbf{C}^T\mathbf{w}_{opt}] \end{aligned} \quad (6.24)$$

The constraints are clearly satisfied by the optimal solution, i.e., $\mathbf{f} - \mathbf{C}^T\mathbf{w}_{opt} = \mathbf{0}$, therefore, we get

$$\Delta \mathbf{w}(k+1) = \mathbf{P}\{\mathbf{I} - \mu \mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\}\Delta \mathbf{w}(k) \quad (6.25)$$

Before continuing we notice that $\mathbf{P}\Delta \mathbf{w}(k) = \Delta \mathbf{w}(k)$. This can be shown for $k \geq 1$ by multiplying Equation (6.25) with \mathbf{P} from the left and using the fact that the matrix \mathbf{P} is idempotent [94], i.e., $\mathbf{P}\mathbf{P} = \mathbf{P}$ and $\mathbf{P}^T = \mathbf{P}$. For $k = 0$ we have $\mathbf{w}(0) = \mathbf{F}$, and

consequently $\Delta \mathbf{w}(0) = \mathbf{F} - \mathbf{w}_{opt}$. Therefore,

$$\begin{aligned}
\mathbf{P}\Delta \mathbf{w}(0) &= \mathbf{P}(\mathbf{F} - \mathbf{w}_{opt}) \\
&= \mathbf{P}\mathbf{F} - \mathbf{P}\mathbf{w}_{opt} \\
&= (\mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T)\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f} - (\mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T)\mathbf{w}_{opt} \\
&= \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{w}_{opt} - \mathbf{w}_{opt} \\
&= \mathbf{F} - \mathbf{w}_{opt}
\end{aligned} \tag{6.26}$$

We can now rewrite Equation (6.25) as

$$\begin{aligned}
\Delta \mathbf{w}(k+1) &= \mathbf{P}\{\mathbf{I} - \mu\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\}\mathbf{P}\Delta \mathbf{w}(k) \\
&= \{\mathbf{P} - \mu\mathbf{P}\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}^T\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\mathbf{P}^T\}\Delta \mathbf{w}(k) \\
&= \{\mathbf{P} - \mu\mathbf{P}\mathbf{P}\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}^T\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\mathbf{P}^T\}\Delta \mathbf{w}(k) \\
&= \{\mathbf{I} - \mu\mathbf{P}\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{P}^T\mathbf{P}\mathbf{X}(k)]^{-1}\mathbf{X}^T(k)\mathbf{P}^T\}\Delta \mathbf{w}(k) \\
&= \{\mathbf{I} - \mu\bar{\mathbf{X}}(k)[\bar{\mathbf{X}}^T(k)\bar{\mathbf{X}}(k)]^{-1}\bar{\mathbf{X}}^T(k)\}\Delta \mathbf{w}(k) \\
&= \mathbf{T}(k)\Delta \mathbf{w}(k)
\end{aligned} \tag{6.27}$$

where $\bar{\mathbf{X}}(k) = \mathbf{P}\mathbf{X}(k)$ and $\mathbf{T}(k) = \{\mathbf{I} - \mu\bar{\mathbf{X}}(k)[\bar{\mathbf{X}}^T(k)\bar{\mathbf{X}}(k)]^{-1}\bar{\mathbf{X}}^T(k)\}$. Usually convergence in the mean can be claimed for difference equations of the same form as Equation (6.27), under the assumption that $E[\Delta \mathbf{w}(k+1)] \propto E[\mathbf{T}(k)]E[\Delta \mathbf{w}(k)]$ and $E[\mathbf{T}(k)]$ being time-invariant with eigenvalues strictly inside the unit circle. To avoid this strong independence assumption we look at the conditions for convergence with probability 1 of the system describing $\Delta \mathbf{w}(k+1)$. In order to guarantee stability of the linear time-variant system of (6.27), consider the following observation

Observation. *The CAP algorithm of Eq. (6.27) is stable, i.e., $\|\Delta \mathbf{w}(k+1)\| \leq \|\Delta \mathbf{w}(k)\|$, for $0 \leq \mu \leq 2$.*

Proof. Using the relation $\|\mathbf{A}\mathbf{B}\|_p \leq \|\mathbf{A}\|_p \cdot \|\mathbf{B}\|_p$ which is valid for all matrix p -norms [94], we get

$$\|\Delta \mathbf{w}(k+1)\|_p = \|\mathbf{T}(k)\Delta \mathbf{w}(k)\|_p \leq \|\mathbf{T}(k)\|_p \cdot \|\Delta \mathbf{w}(k)\|_p = \|\Delta \mathbf{w}(k)\|_p \tag{6.28}$$

where we used $\|\mathbf{T}(k)\|_p = 1$ for $0 \leq \mu \leq 2$. ■

As for the asymptotic stability we state the following theorem

Theorem 2. *If the input signal is persistently exciting, then the solution of (6.27) and, consequently, the CAP algorithm is asymptotically stable for $0 < \mu < 2$.*

Proof. Using the SVD we can write the transformed input matrix $\bar{\mathbf{X}}(k) = \mathbf{P}\mathbf{X}(k)$ as $\bar{\mathbf{X}}(k) = \mathbf{U}(k)\Sigma(k)\mathbf{V}^T(k)$, where the unitary matrices $\mathbf{U}(k) \in \mathbb{R}^{N \times N}$ and $\mathbf{V}(k) \in \mathbb{R}^{P \times P}$ contain the left and right *singular vectors*, respectively, and $\Sigma(k) \in \mathbb{R}^{N \times P}$ contains the *singular values* on its main diagonal. Consequently we have

$$\begin{aligned} \Delta \mathbf{w}(k+1) &= \mathbf{T}(k)\Delta \mathbf{w}(k) \\ &= \left\{ \mathbf{I} - \mu \mathbf{U}(k)\Sigma(k)\mathbf{V}^T(k) \left[\mathbf{V}(k)\Sigma^T(k)\Sigma(k)\mathbf{V}^T(k) \right]^{-1} \mathbf{V}(k)\Sigma^T(k)\mathbf{U}^T(k) \right\} \Delta \mathbf{w}(k) \\ &= \left\{ \mathbf{I} - \mu \mathbf{U}(k)\Sigma(k) \left[\Sigma^T(k)\Sigma(k) \right]^{-1} \Sigma^T(k)\mathbf{U}^T(k) \right\} \Delta \mathbf{w}(k) \end{aligned} \quad (6.29)$$

where we used the fact that for two invertible matrices \mathbf{A} and \mathbf{B} , $(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$. The matrix $\Sigma(k) \left[\Sigma^T(k)\Sigma(k) \right]^{-1} \Sigma^T(k) \in \mathbb{R}^{N \times N}$ is a diagonal matrix with P ones and $N - P$ zeros, i.e.,

$$\Sigma(k) \left[\Sigma^T(k)\Sigma(k) \right]^{-1} \Sigma^T(k) = \begin{pmatrix} \mathbf{I}_{P \times P} & \mathbf{0}_{N-P \times 1}^T \\ \mathbf{0}_{N-P \times 1} & \mathbf{0}_{N-P \times N-P} \end{pmatrix} \quad (6.30)$$

Therefore,

$$\begin{aligned} \|\Delta \mathbf{w}(k+1)\|^2 &= \Delta \mathbf{w}^T(k)\mathbf{T}^T(k)\mathbf{T}(k)\Delta \mathbf{w}(k) \\ &= \Delta \mathbf{w}^T(k) \left\{ \mathbf{I} - \mu(2 - \mu)\mathbf{U}(k)\Sigma(k) \left[\Sigma^T(k)\Sigma(k) \right]^{-1} \Sigma^T(k)\mathbf{U}^T(k) \right\} \Delta \mathbf{w}(k) \\ &= \|\Delta \mathbf{w}(k)\|^2 - \mu(2 - \mu)\Delta \tilde{\mathbf{w}}^T(k)\Sigma(k) \left[\Sigma^T(k)\Sigma(k) \right]^{-1} \Sigma^T(k)\Delta \tilde{\mathbf{w}}(k) \\ &= \|\Delta \mathbf{w}(k)\|^2 - \mu(2 - \mu) \sum_{i=1}^P \Delta \tilde{w}_i^2 \end{aligned} \quad (6.31)$$

where $\Delta \tilde{\mathbf{w}}(k) = \mathbf{U}^T(k)\Delta \mathbf{w}(k)$. For the asymptotic stability, we can conclude that $\|\Delta \mathbf{w}(k+1)\|^2$ remains constant, i.e., $\|\Delta \mathbf{w}(k+1)\|^2 = \|\Delta \mathbf{w}(k)\|^2$, during an interval $[k_1, k_2]$ if and only if we choose $\mu = 2$ or $\mu = 0$, or $\Delta \mathbf{w}(k)$ is orthogonal to the P left singular

vectors of $\bar{\mathbf{X}}(k) = \mathbf{P}\mathbf{X}(k)$ corresponding to the P non-zero singular values in $\Sigma(k)$ for all $k \in [k_1, k_2]$, i.e, $\mathbf{U}^T(k)\Delta\mathbf{w}(k) = [\underbrace{0, \dots, 0}_P, \underbrace{*, \dots, *}_{N-P+1}]^T, \forall k \in [k_1, k_2]$, where the elements denoted $*$ can take arbitrary values. However, if the input signal is persistently exciting, we can define an infinite number of sets $\mathcal{S}_i = \{\bar{\mathbf{U}}_{k_{1i}}, \dots, \bar{\mathbf{U}}_{k_{2i}}\}$, where $\bar{\mathbf{U}}(k) \in \mathbb{R}^{N \times P}$ denotes the P first columns of $\mathbf{U}(k)$, with $M \leq (k_{2i} - k_{1i}) \leq M'$, such that each set \mathcal{S}_i completely spans \mathbb{R}^N for some finite value of $M' > 0$. This makes it impossible to have $\Delta\mathbf{w}(k)$ orthogonal to all $\bar{\mathbf{U}}(k) \in \mathcal{S}_i$ and, as a consequence, $\|\Delta\mathbf{w}_{k_{2i}}\|^2 < \|\Delta\mathbf{w}_{k_{1i}}\|^2$. Since the number of sets is infinite, $\|\Delta\mathbf{w}(k+1)\|^2 \rightarrow 0$ for $k \rightarrow \infty$ [43], which concludes the proof. \blacksquare

6.6 Simulation Results

6.6.1 Identification of Plant with Linear Phase

A first experiment was carried out in a system identification problem where the filter coefficients were constrained to preserve linear phase at every iteration. For this example we chose $N = 10$ and, in order to meet the linear phase requirement, we made

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{N/2} \\ \mathbf{0}^T \\ -\mathbf{J}_{N/2} \end{bmatrix} \quad (6.32)$$

with \mathbf{J} being a reversal matrix (an identity matrix with all rows in reversed order), and

$$\mathbf{f} = [0 \dots 0]^T \quad (6.33)$$

This setup was employed to show the improvement of the convergence speed when the number of data-reuses P is increased. The input signal consists of colored noise with a zero mean and unity variance with eigenvalue spread around 2000, and the reference signal was obtained after filtering the input by a linear-phase FIR filter and adding observation noise with variance equal to 10^{-10} .

Fig. 6.3 shows the learning curves for the CAP, the HCAP, and the SM-CAP algorithms for $P = 1$, $P = 2$, and $P = 4$. The CAP and HCAP algorithms present identical results in

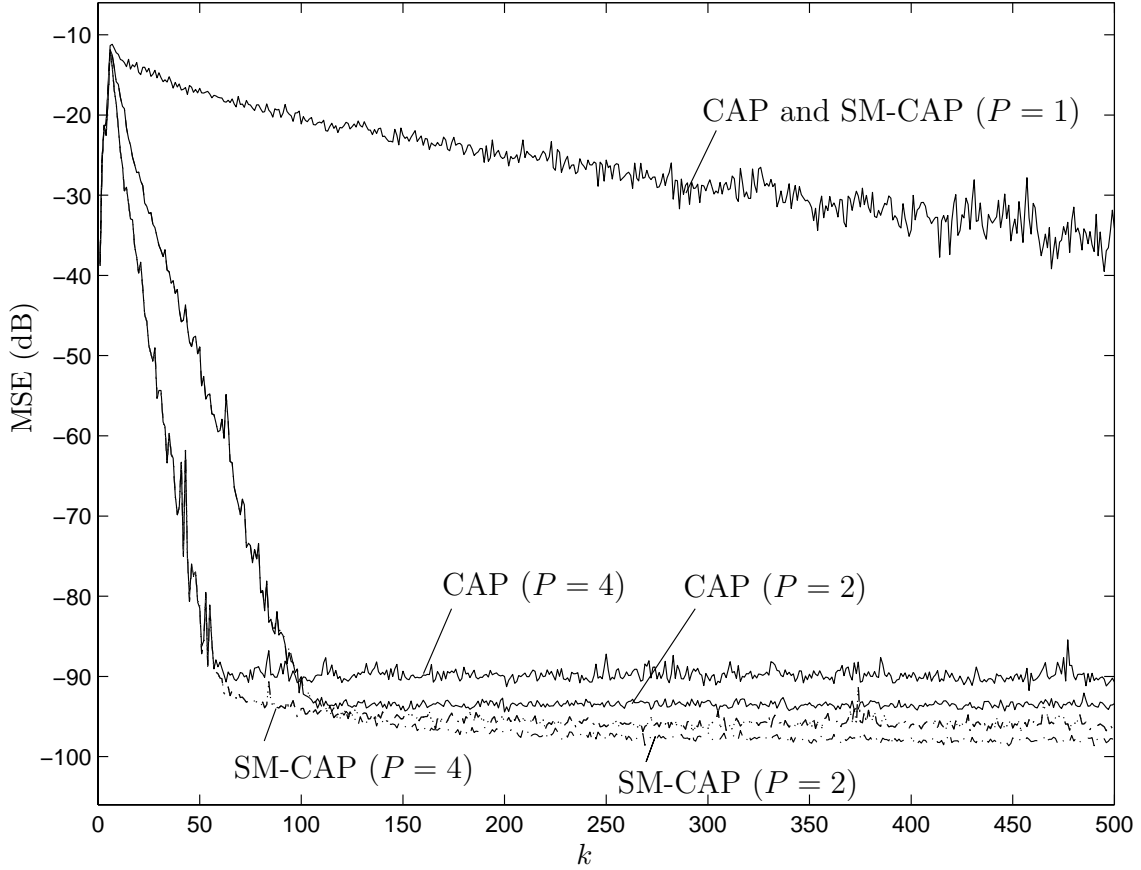


Figure 6.3: Learning curves for the CAP and the SM-CAP algorithms with $P = 1$, $P = 2$, and $P = 4$ data reuses, $\sigma_n^2 = 10^{-10}$, $\gamma = 3\sigma_n$, and colored input signal.

infinite precision environment. The value of γ in the SM-CAP algorithm was chosen equal to $\sqrt{6}\sigma_n$. A higher value would result in less frequent updates but in a slightly higher final misadjustment. It is clear from this figure that for the CAP algorithm the misadjustment increases with P . It is also clear from this figure that the misadjustment with the SM-CAP algorithm is lower than with the CAP algorithm, and that the misadjustment increases more slowly when P is increased. The only way for the CAP algorithm to achieve the low misadjustment of the SM-CAP is through the introduction of a step size resulting in a slower convergence. Furthermore, in 500 iterations the SM-CAP algorithm performed updates in 485, 111, and 100 time instants for $P = 1$, $P = 2$, and $P = 4$, respectively. In other words, the SM-CAP algorithm with $P = 4$ had a better performance than the CAP algorithm while performing updates for only a fraction of data.

Also for this first experiment, Fig. 6.4 shows that we have no bias in the coefficient vector after convergence. In this experiment the CAP algorithm and the SM-CAP algorithm presented identical bias curves. It is worth mentioning that the optimum coefficient vector used to compute the coefficient error vector was obtained from (6.21) after replacing $\mathbf{R}^{-1}\mathbf{p}$ (the Wiener solution) by \mathbf{w}_{us} (the FIR unknown system). The input signal was taken as colored noise generated by filtering white noise through a filter with a pole at $\alpha = 0.099$. The autocorrelation matrix for this example is given by

$$\mathbf{R} = \frac{\sigma_{WGN}^2}{1 - \alpha^2} \begin{bmatrix} 1 & -\alpha & (-\alpha)^2 & \cdots & (-\alpha)^{N-1} \\ -\alpha & 1 & -\alpha & \cdots & (-\alpha)^{N-2} \\ \vdots & \vdots & \ddots & \vdots & \\ (-\alpha)^{N-1} & (-\alpha)^{N-2} & \cdots & & 1 \end{bmatrix} \quad (6.34)$$

where σ_{WGN}^2 is set such that $\frac{\sigma_{WGN}^2}{1 - \alpha^2}$ corresponds to the desired input signal variance σ_x^2 .

6.6.2 Linearly-Constrained Minimum-Variance

Filtering of Sinusoids

A second experiment was done where the received signal consists of three sinusoids in white noise:

$$x(k) = \sin(0.3k\pi) + \sin(0.325k\pi) + \sin(0.7k\pi) + n(k) \quad (6.35)$$

where $n(k)$ is white noise with power such that the SNR is 40 dB. The filter is constrained to pass frequency components of 0.1rad/s and 0.25rad/s undistorted which results in the following constraint matrix and vector:

$$\mathbf{C}^T = \begin{bmatrix} 1 & \cos(0.2\pi) & \cdots & \cos[(N-1)0.2\pi] \\ 1 & \cos(0.5\pi) & \cdots & \cos[(N-1)0.5\pi] \\ 0 & \sin(0.2\pi) & \cdots & \sin[(N-1)0.2\pi] \\ 0 & \sin(0.5\pi) & \cdots & \sin[(N-1)0.5\pi] \end{bmatrix} \quad (6.36)$$

$$\mathbf{f}^T = [1 \ 1 \ 0 \ 0] \quad (6.37)$$

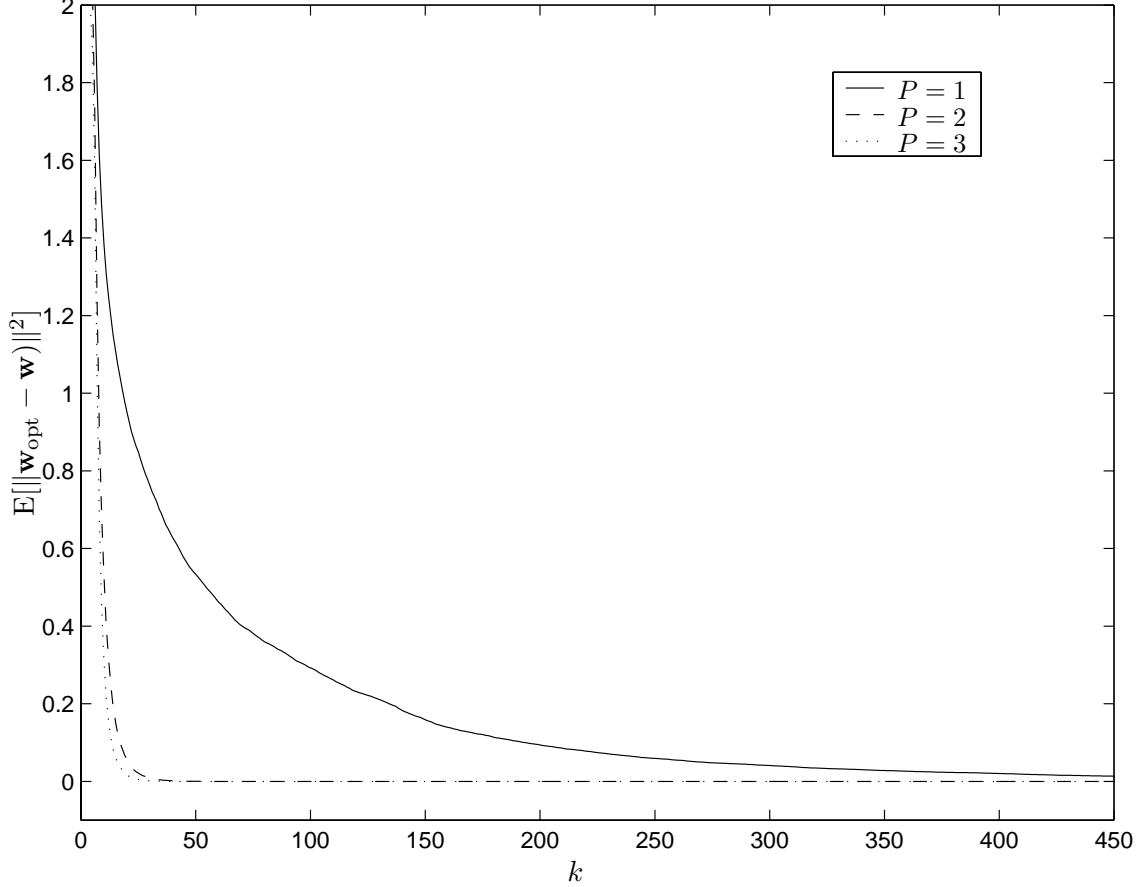


Figure 6.4: First experiment: no bias in the coefficient vector.

In this example the reference signal is set to zero, i.e., $e(k) = -\mathbf{x}^T(k)\mathbf{w}(k)$.

The mean output energy (MOE) is shown in Fig. 6.5 for the CAP and the SM-CAP algorithms for $P = 3$. The threshold γ was set to $4\sigma_n$. A step size $\mu_{CAP} = 0.15$ was used with the CAP to obtain a steady-state close to the SM-CAP algorithm. We see from the figure that the SM-CAP curve is less noisy than the CAP curve during the initial 1500 iterations. After the convergence both algorithm have similar steady-state value. In 5000 iterations, the average number of updates for the SM-CAP algorithm was 790 as compared with 5000 updates for the CAP algorithm. The norm of the coefficient-error vector for values of P from 1 to 3 is depicted in Fig. 6.6. The optimum coefficient vector in this case was also obtained from (2.9) and computing \mathbf{R} with

$$E[x(k)x(k-i)] = \frac{1}{2}[\cos(0.3\pi i) + \cos(0.325\pi i) + \cos(0.7\pi i)] + \sigma_n^2\delta_i \quad (6.38)$$

From Fig. 6.6 we can realize that, although faster, the CAP (or HCAP) and the SM-CAP

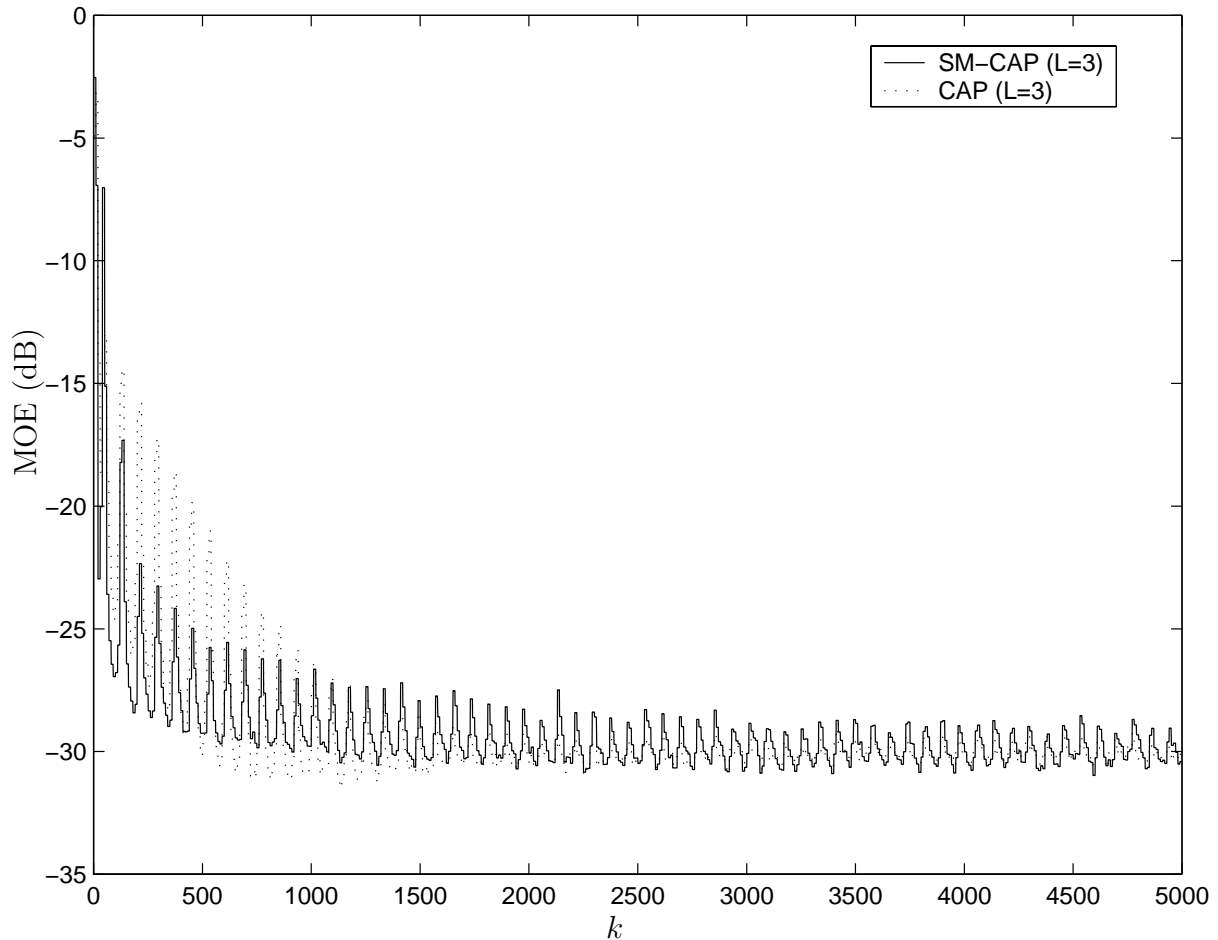


Figure 6.5: The mean output power

algorithms present, for this particular experiment, an increasing misadjustment with P , specially when this number of projections is higher than 2 (this value corresponding to the BNDRLMS algorithm for the CAP algorithm).

6.6.3 Interference Suppression in a CDMA Communications System

In this section, we apply the constrained adaptive algorithms to the case of single-user detection in DS-CDMA mobile communications systems. The goal of this example is to demonstrate the effect of the correction term used in the SM-CAP and the CAP algorithms, when the algorithms operate in finite-precision. Using a similar setup as in [71] we can

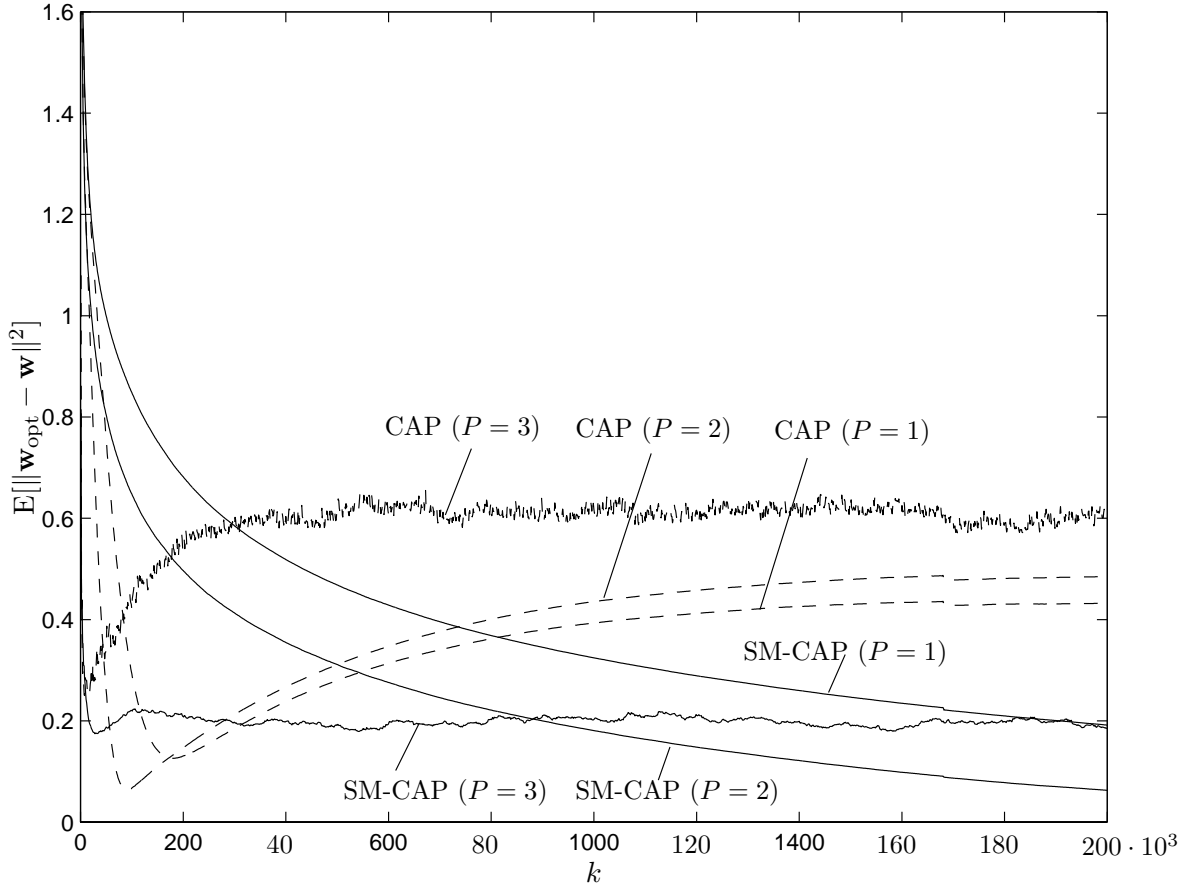


Figure 6.6: Coefficient-vector deviation for the second experiment.

compare our results with those of the APOC-LMS algorithm [71] which, as was noted earlier in this chapter, does not make use of a correction term.

The received signal for a system with K simultaneous users can be written as

$$\mathbf{x}(k) = \sum_{i=1}^K A_i b_i(k) \mathbf{s}_i + \mathbf{n}(k) \quad (6.39)$$

where for the i th user, A_i is the amplitude, $\mathbf{s}_i \in \mathbb{R}^N$ is the spreading code, and $b_i(k) \in \{\pm 1\}$ is the transmitted bit. In the case of single-user detection, we are only interested in detecting one user (here assumed to be $i = 1$). One way to construct the receiver coefficients is to minimize the Euclidean distance of the coefficient update under the constraint that the desired user's spreading code can pass with unity response, i.e.,

$$\mathbf{w}(k+1) = \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to } \mathbf{s}_1^T \mathbf{w} = 1 \quad (6.40)$$

where, using the notation of this chapter, we see that the reference signal $d(k) = 0$, $\mathbf{C} = \mathbf{s}_1$,

and $\mathbf{f} = 1$. To solve this problem adaptively, we can apply the CAP algorithm with one data-reuse, i.e., $P = 1$.

For the SMF approach proposed in [71], it was suggested that the receiver coefficients should be chosen such that they belong to the hyperplane defined by $\mathbf{C}^T \mathbf{w} = 1$ and also to the constraint set defined by

$$\mathcal{H}(k) = \{\mathbf{w} \in \mathbb{R}^N : |\mathbf{w}^T \mathbf{x}(k)| \leq A_1 + \gamma'\}.$$

Consequently, we can apply the SM-CAP with $P = 1$ data-reuse and $\gamma = A_1 + \gamma'$ to implement the single-user detector.

The system considered contained $K = 10$ users. The spreading codes for each users were taken as random unit-norm vector, where the interfering user codes were changed for each of the $M = 500$ realizations. The signal-to-noise ratio (SNR) for the desired user was set to 20dB, and the interfering-users amplitudes were set to 5 times the desired user amplitude. The signal-to-interference plus noise ratio (SINR) versus the iterations was measured for all algorithms using

$$\text{SINR}(k) = \frac{\sum_{i=1}^M [\mathbf{w}^T(k) A_1 \mathbf{s}_1]^2}{\sum_{i=1}^M [\mathbf{w}^T(k) \{\mathbf{x}(k) - A_1 b_1(k) \mathbf{s}_1\}]^2}$$

The CAP ($P = 1$), SM-CAP ($P = 1$), NCLMS without correction term, and APOC-LMS [71] algorithms were implemented using 16-bits fixed-point arithmetic. For the SMF algorithms we used $\gamma = A_1 + \gamma' = A_1 + 0.1$ according to the choice made in [71].

Figure 6.7 shows the SINR versus the iteration k . We see from the figure that SM-CAP and APOC-LMS algorithms have a similar performance, converging faster to a steady-state solution than the other algorithms. In 1000 iterations the number of times an update took place for SM-CAP and APOC-LMS were 300. The value of γ' will trade off the number of required updates with a slight decrease in SINR steady-state value.

Figure 6.8 shows the deviation from the constraint hyperplane versus the iterations. Here we can see that the CAP and SM-CAP algorithms will not depart from the constraint plane. However, the NCLMS without correction term and APOC-NLMS will depart from the constrain plane due to accumulation of round-off errors. For a continuous operation or very long data records, algorithms lacking a mechanism to enforce the constraints may

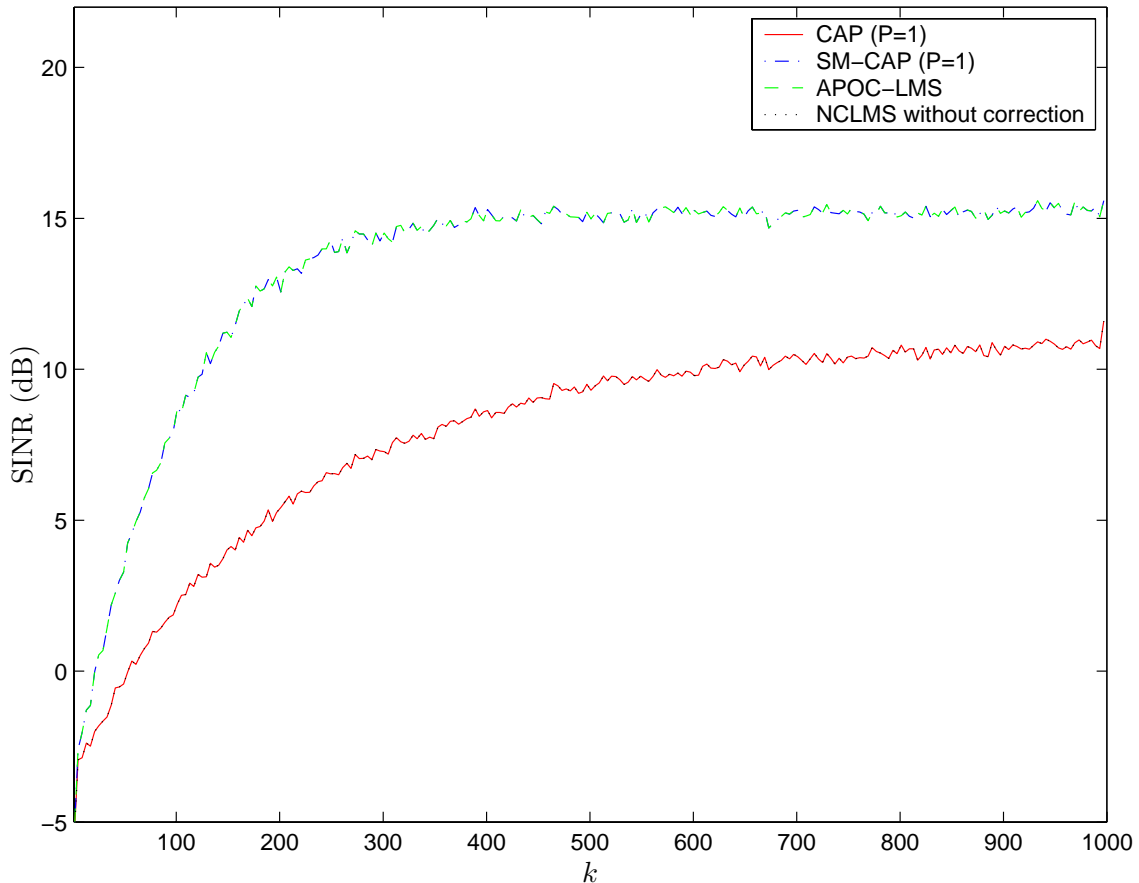


Figure 6.7: SINR versus iteration for the CAP ($P = 1$), SM-CAP ($P = 1$), APOC-LMS, NCLMS (without correction term) algorithms, 16-bits fixed-point arithmetic, and $\gamma = (1 + 0.1)$.

not be a suitable choice [85]. One solution for these last type of projection algorithms would be to periodically project the solution back onto the hyperplane spanned by the constraints [25]. We note that the departure from the constraint plane is slower for the APOC-LMS algorithm as compared to the NCLMS without correction term. This is due to the sparse update in time coming from the SMF approach to adaptive filtering.

6.7 Conclusions

In this chapter, we have introduced the constrained version of the affine-projection algorithm as well as an efficient Householder-transform constrained version. We also derived

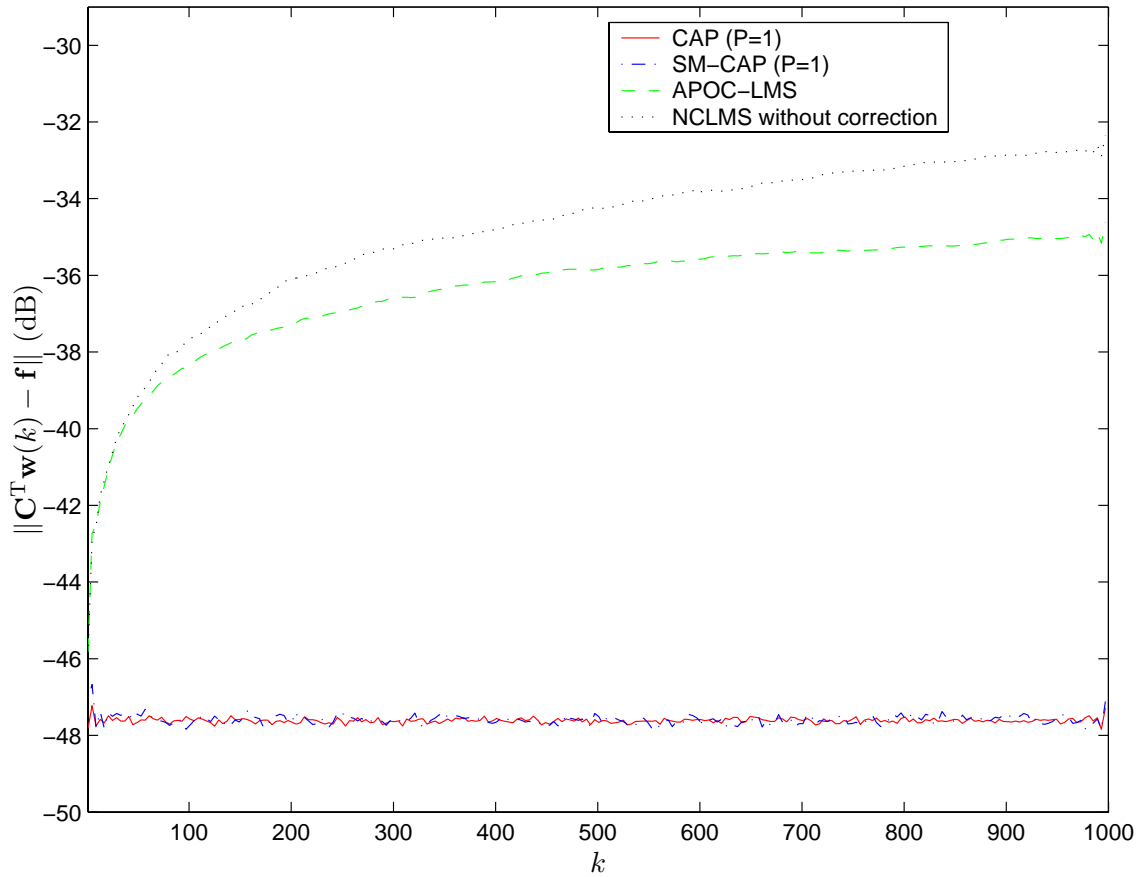


Figure 6.8: Constraint deviation versus iteration for the CAP ($P = 1$), SM-CAP ($P = 1$), APOC-LMS, NCLMS (without correction term) algorithms, for 16-bits fixed-point arithmetic, and $\gamma = (1 + 0.1)$.

a data-selective version of the constrained affine-projection algorithm that can in certain applications substantially reduce the number of required updates. The simulation results confirmed that the proposed algorithms leads to fast convergence speed, low misadjustment, and reduction in the number of updates. The analysis claims of an unbiased solution was supported by a system-identification example, where the filter was constrained to have linear phase.

Chapter 7

Partial-Update NLMS Algorithms with Data-Selective Updating

In this chapter, we introduce and present mean-squared convergence analysis for the partial-update normalized least-mean square (PU-NLMS) algorithm with closed-form expressions for the case of white input signals. The formulas presented here are more accurate than the ones found in the literature for the PU-NLMS algorithm. Thereafter, the ideas of the partial-update NLMS-type algorithms found in the literature are incorporated in the framework of set-membership filtering, from which data-selective NLMS-type of algorithms with partial-update are derived. The new algorithms, referred to herein as the set-membership partial-update normalized least-mean square (SM-PU-NLMS) algorithms, combine the data-selective updating from set-membership filtering with the reduced computational complexity from partial updating. A thorough discussion of the SM-PU-NLMS algorithms follows, where we propose different update strategies, provide stability analysis, and closed-form formulas for excess mean-squared error (MSE). Simulation results verify the analysis for the PU-NLMS algorithm and the good performance of the SM-PU-NLMS algorithms in terms of convergence speed, final misadjustment, and reduced computational complexity.

7.1 Introduction

When implementing an adaptive-filtering algorithm, the affordable number of coefficients that can be used will depend on the application in question, the adaptation algorithm, and the hardware chosen for implementation. With the choice of algorithms ranging from the simple least-mean square (LMS) algorithm to the more complex recursive least squares (RLS) algorithm, tradeoffs between performance criteria such as, e.g., computational complexity and convergence rate, have to be made. In certain applications, the use of the RLS algorithm is prohibitive due to the high computational complexity and in such cases we must resort to simpler algorithms. As an example, consider the acoustic echo cancellation application where the adaptive filter may require thousands of coefficients [111]. This large number of filter coefficients may impair even the implementation of low computational complexity algorithms, such as the normalized least-mean square (NLMS) algorithm [1]. As an alternative, instead of reducing filter order, one may choose to *update only part of the filter coefficient vector at each time instant*. Such algorithms, referred to as partial-update (PU) algorithms, can reduce computational complexity while performing close to their full-update counterparts in terms of convergence rate and final mean-squared error (MSE). In the literature one can find several variants of the LMS and the NLMS algorithms with partial updates [73, 74, 75, 114, 115, 116, 117, 118, 119], as well as more computationally complex variants based on the affine projection algorithm [14].

The objective of this chapter is to propose a framework which combines set-membership filtering with partial-update. The resulting algorithms benefit from the data-selective updating related to the set-membership framework reducing the average computational complexity, and also from the reduced computational complexity obtained with the partial update of the coefficient vector. The main contributions are the development of updating schemes that guarantee performance comparable to that of set-membership filtering algorithms and partial-updating algorithms whereas computational complexity is reduced with respect to both updating schemes. Furthermore, a thorough discussion on the properties of the developed algorithms is presented stating their most important features and contributing to improve the understanding of their behavior.

When presenting the basis for developing the new algorithm, here referred to as set-membership partial-update NLMS (SM-PU-NLMS) algorithm, we rederive and analyze one particular case of the partial-update NLMS (PU-NLMS) algorithm introduced in [14, 75] that obeys the principle of minimum disturbance [14]. The results from our analysis, which is based on order statistics, yield more accurate bounds on step size and on the prediction of excess MSE when compared to the results presented in [14]. We also clarify the relationship between the PU-NLMS and M-Max NLMS [74, 116] algorithms, whereby we show that the M-Max NLMS algorithm uses an instantaneous estimate of the step size that achieves the fastest convergence in the MSE.

We propose two versions of the SM-PU-NLMS algorithm: one version updates a constant number of coefficients whenever an update is required, whereas the other version allows the number of coefficients to be updated vary up to a maximum pre-defined number. In both versions the SMF criterion is used in tandem with the PU criterion to construct guidelines that will determine when and which coefficients shall be updated. We also provide proof of convergence for the SM-PU-NLMS algorithm in the mean-squared sense in the case of white input sequences.

The organization of the chapter is as follows. Section 7.2 reviews the PU-NLMS algorithm for the particular case where the coefficients to be updated are not contiguous and are chosen based on the minimum disturbance criterion. We also provide an analysis in the mean-squared sense that is novel for this algorithm and allows new insights to its behavior. Section 7.3 contains the derivation of the new algorithm. Section 3 also provides discussion on the convergence properties of the new algorithm. Section 7.4 discusses computational complexity issues of the algorithms, followed by simulations in Section 7.5. In this section we validate our analysis of the PU-NLMS algorithm and compare our results with those available in the literature. We also compare the SM-PU-NLMS algorithm for different choices of update strategy and evaluate the reduction in the computational complexity resulting from the combination of partial-update and set-membership approaches. Conclusions are given in Section 7.6.

7.2 The Partial-Update NLMS Algorithm

This section reviews the partial-update NLMS (PU-NLMS) algorithm proposed in [14, 75]. The approach taken here is slightly different from that in [14, 75], but the final algorithm is the same as the one that satisfies the minimum disturbance criterion. We also provide analysis in the mean-squared sense with new bounds on the step size to be used in the PU-NLMS algorithm that are more accurate than the one given in [14].

The objective in PU adaptation is to derive an algorithm that only updates L out of the N filter coefficients. Let the L coefficients to be updated at time instant k be specified by an index set $\mathcal{I}_L(k) = \{i_1(k), \dots, i_L(k)\}$ with $\{i_j(k)\}_{j=1}^L$ taken from the set $\{1, \dots, N\}$. Note that $\mathcal{I}_L(k)$ depends on the time instant k . As a consequence, the L coefficients to be updated can change between consecutive time instants. A question that naturally arises is “Which L coefficients should be updated?” The answer can be related to the optimization criterion chosen for the algorithm derivation.

In the conventional NLMS algorithm, the new coefficient vector can be obtained as the vector $\mathbf{w}(k+1)$ that minimizes the Euclidean distance $\|\mathbf{w} - \mathbf{w}(k)\|^2$ subject to the constraint of zero *a posteriori* error. Applying the same idea for the partial update of vector $\mathbf{w}(k)$, we take the updated vector $\mathbf{w}(k+1)$ as the vector minimizing the Euclidean distance $\|\mathbf{w} - \mathbf{w}(k)\|^2$ subject to the constraint of zero *a posteriori* error *with the additional constraint of updating only L coefficients*. For this purpose, we introduce the diagonal matrix $\mathbf{A}_{\mathcal{I}_L(k)}$ having L elements equal to one in the positions indicated by $\mathcal{I}_L(k)$ and zeros elsewhere. Defining the complementary matrix $\tilde{\mathbf{A}}_{\mathcal{I}_L(k)} = \mathbf{I} - \mathbf{A}_{\mathcal{I}_L(k)}$ will give $\tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \mathbf{w}(k+1) = \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \mathbf{w}(k)$, which means that only L coefficients are updated. With this notation the optimization criterion for the partial update can be formulated as

$$\mathbf{w}(k+1) = \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to} \quad (7.1)$$

$$\begin{cases} \mathbf{x}^T(k) \mathbf{w} = d(k) \\ \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} [\mathbf{w} - \mathbf{w}(k)] = \mathbf{0} \end{cases}$$

Applying the method of Lagrange multipliers (see Appendix A7.1 of this chapter) gives

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{e(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}. \quad (7.2)$$

We see from (7.2) that only the coefficients of $\mathbf{w}(k)$ indicated by the index set $\mathcal{I}_L(k)$ are updated, whereas the remaining coefficients are not changed from iteration k to iteration $k+1$.

We now concentrate on the choice of the index set $\mathcal{I}_L(k)$. Substituting the recursions in (7.2) into (7.1) we get the Euclidean distance as

$$E(k) = \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 = \left\| \frac{e(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \right\|^2 = \frac{1}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} e^2(k) \quad (7.3)$$

For a given value of $e^2(k)$, we can conclude that $E(k)$ achieves its minimum when $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|$ is maximized. In other words, we should update the L coefficients of $\mathbf{w}(k)$ related to the elements of $\mathbf{x}(k)$ with the largest norm.

In order to control stability, convergence speed, and error in the mean-squared sense a step size is required, leading to the following final recursion for the PU-NLMS algorithm

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \frac{e(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \quad (7.4)$$

The bound on the step size is given by (see Appendix A7.2 of this chapter)

$$0 < \mu < \frac{2}{\mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right]} \approx \frac{2\mathbb{E} [\tilde{r}^2(k)]}{N\sigma_x^2} \quad (7.5)$$

where $\tilde{r}^2(k)$ has the same distribution as $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2$, and $r^2(k)$ has the same distribution as $\|\mathbf{x}(k)\|^2$, which in this particular case is a sample of an independent process with chi-distribution with N degrees of freedom, $\mathbb{E} [r^2(k)] = N\sigma_x^2$. For given N and L , $\mathbb{E} [\tilde{r}^2(k)]$ can be evaluated numerically, as shown in Appendix A7.3 of this chapter. It can also be shown that $L\sigma_x^2 \leq \mathbb{E} [\tilde{r}^2(k)] \leq N\sigma_x^2$ for white Gaussian input signals (see Lemma 3 in Appendix A7.3 of this chapter). A more pessimistic bound on the step size, $0 \leq \mu \leq 2L/N$, was given in [14] as a consequence of the approximation $\mathbb{E} [\tilde{r}^2(k)] \approx L\sigma_x^2$.

In Appendix A7.2 of this chapter it is shown that if order statistics is used, the final

excess MSE after convergence is given by

$$\begin{aligned}\Delta\xi_{exc} &\approx N \frac{\mu\sigma_n^2\sigma_x^2}{2 - \mu\mathbb{E}\left[\frac{r^2(k)}{\tilde{r}^2(k)}\right]} \mathbb{E}\left[\frac{1}{\tilde{r}^2(k)}\right] \\ &\approx N \frac{\mu\sigma_n^2\sigma_x^2}{2\mathbb{E}[\tilde{r}^2(k)] - \mu N\sigma_x^2}\end{aligned}\tag{7.6}$$

When $L = N$, Equation (7.6) is consistent with the results obtained for the conventional NLMS algorithm in [44]. The algorithm presented in this section is identical to the partial-update NLMS algorithm with multiple blocks of contiguous coefficients to be updated proposed in [14] for the case of unity block size and L blocks. Choosing blocks of filter coefficients rather than the L coefficients corresponding to the elements with the largest magnitude in the input-signal vector can reduce the amount of memory required for implementation [118]. However, such an approach will no longer perform an update that minimizes the criterion in (7.1), resulting in slower convergence speed.

For a step size $\mu(k) = \bar{\mu}\|\mathbf{A}_{\mathcal{I}_{L(k)}}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$, the PU-NLMS in (7.4) becomes identical to the M-Max NLMS algorithm of [74]. For $\bar{\mu} = 1$, the solution is the projection of the solution of the NLMS algorithm with unity step size onto the direction of $\mathbf{A}_{\mathcal{I}_{L(k)}}\mathbf{x}(k)$, as illustrated in Figure 7.1. Furthermore, $\mu = \|\mathbf{A}_{\mathcal{I}_{L(k)}}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$ corresponds to the instantaneous estimate of $\mathbb{E}[r^2(k)/\tilde{r}^2(k)]$ which gives the fastest convergence, as observed in Appendix A7.2 of this chapter.

7.3 The Set-Membership Partial-Update NLMS Algorithm

In this section we merge the ideas of partial updating and set-membership filtering. The goal is to combine the advantages of SMF and PU in order to obtain an algorithm with sparse updating and low computational complexity per update. The following subsections present the algorithm derivation and discuss convergence issues.

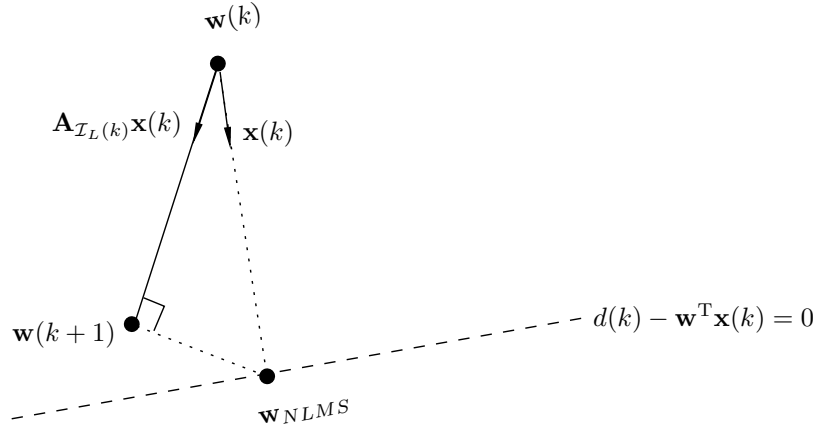


Figure 7.1: The solution $\mathbf{w}(k+1)$ is the PU-NLMS algorithm update obtained with a time-varying step size $\mu(k) = \|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$, or equivalently, the M-Max NLMS algorithm with unity step size.

7.3.1 Algorithm Derivation

Our approach is to seek a coefficient vector that minimizes the Euclidean distance $\|\mathbf{w} - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w} \in \mathcal{H}(k)$ with the additional constraint of updating only L coefficients. This means that if $\mathbf{w}(k) \in \mathcal{H}(k)$, the minimum distance is zero and no update is required. However, when $\mathbf{w}(k) \notin \mathcal{H}(k)$, the new update is obtained as the solution to the following optimization problem

$$\mathbf{w}(k+1) = \arg \min_{\mathbf{w}} \|\mathbf{w} - \mathbf{w}(k)\|^2 \text{ subject to:} \quad (7.7)$$

$$\begin{cases} d(k) - \mathbf{x}^T(k)\mathbf{w} = g(k) \\ \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} [\mathbf{w} - \mathbf{w}(k)] = \mathbf{0} \end{cases}$$

where $g(k)$ is a parameter that determines a point within the constraint set $\mathcal{H}(k)$, or equivalently, $|g(k)| \leq \gamma$. Here $g(k)$ is chosen such that the updated vector belongs to the closest bounding hyperplane in $\mathcal{H}(k)$, i.e., $g(k) = \gamma e(k)/|e(k)|$. The updating equation is obtained in a similar manner as the PU-NLMS algorithm in the previous section

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k) \frac{e(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \quad (7.8)$$

but here the step size $\alpha(k)$ is data dependent and given by

$$\alpha(k) = \begin{cases} 1 - \gamma/|e(k)| & \text{when } \mathbf{w}(k) \notin \mathcal{H}(k), \text{ i.e., if } |e(k)| > \gamma \\ 0 & \text{otherwise} \end{cases} \quad (7.9)$$

The index set $\mathcal{I}_L(k)$ specifying the coefficients to be updated is chosen as in the PU-NLMS algorithm, i.e., the L coefficients in the input vector $\mathbf{x}(k)$ having the largest norm. The algorithm is similar in form to the PU-NLMS algorithm [14], but not in philosophy or in derivation.

A geometrical interpretation of the SM-PU-NLMS algorithm update is given in Figure 7.2 for the case of $N = 3$ filter coefficients and $L = 1$ coefficient to be updated. In the figure, the component $x(k-2)$ is the element of largest magnitude in $\mathbf{x}(k)$, therefore the matrix $\mathbf{A}_{\mathcal{I}_{L_3}(k)}$, which specifies the coefficients to update in $\mathbf{w}(k)$, is equal to $\mathbf{A}_{\mathcal{I}_{L_3}(k)} = \text{diag}(0 \ 0 \ 1)$. The solution \mathbf{w}^\perp in Figure 7.2 is the solution obtained by the SM-NLMS algorithm abiding the orthogonality principle. The angle θ shown in Figure 7.2 denotes the angle between the direction of update $\mathbf{A}_{\mathcal{I}_{L_3}(k)} \mathbf{x}(k) = [0 \ 0 \ x(k-2)]^T$ and the input vector $\mathbf{x}(k)$, and is given from standard vector algebra by the relation $\cos \theta = \frac{|x(k-2)|}{\sqrt{|x(k)|^2 + |x(k-1)|^2 + |x(k-2)|^2}}$. In the general case, with L coefficients in the update, the angle θ in \mathbb{R}^N is given by $\cos \theta = \frac{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|}{\|\mathbf{x}(k)\|}$.

In order to take the solution of the SM-PU-NLMS algorithm closer to the orthogonal projection than the solution, \mathbf{w}^\perp , before the update, consider the bound given by the following lemma:

Lemma 3. $\|\mathbf{w}(k+1) - \mathbf{w}^\perp\|^2 \leq \|\mathbf{w}^\perp - \mathbf{w}(k)\|^2$ for $\frac{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2}{\|\mathbf{x}(k)\|^2} \geq \frac{1}{2}$.

Proof. The orthogonal projection is given by $\mathbf{w}^\perp = \mathbf{w}(k) + \alpha(k) \frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2}$ [69] where $\alpha(k)$ is given by Equation (7.9). Consequently, $\|\mathbf{w}^\perp - \mathbf{w}(k)\|^2 = \frac{\alpha^2(k)e^2(k)}{\|\mathbf{x}(k)\|^2}$. Since $\mathbf{w}(k+1)$ and \mathbf{w}^\perp lie in the same hyperplane, we have $[\mathbf{w}(k+1) - \mathbf{w}^\perp] \perp [\mathbf{w}^\perp - \mathbf{w}(k)]$. Therefore, $\|\mathbf{w}(k+1) - \mathbf{w}^\perp\|^2 = \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 - \|\mathbf{w}^\perp - \mathbf{w}(k)\|^2 = \frac{\alpha^2(k)e^2(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} - \frac{\alpha^2(k)e^2(k)}{\|\mathbf{x}(k)\|^2}$. For $\|\mathbf{w}(k+1) - \mathbf{w}^\perp\|^2 \leq \|\mathbf{w}^\perp - \mathbf{w}(k)\|^2$ to hold $\frac{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2}{\|\mathbf{x}(k)\|^2} \geq \frac{1}{2}$ is required. ■

The lemma tells us that if the instantaneous power in the input vector corresponding to the partial update is larger than half of the total instantaneous power, the SM-PU-NLMS

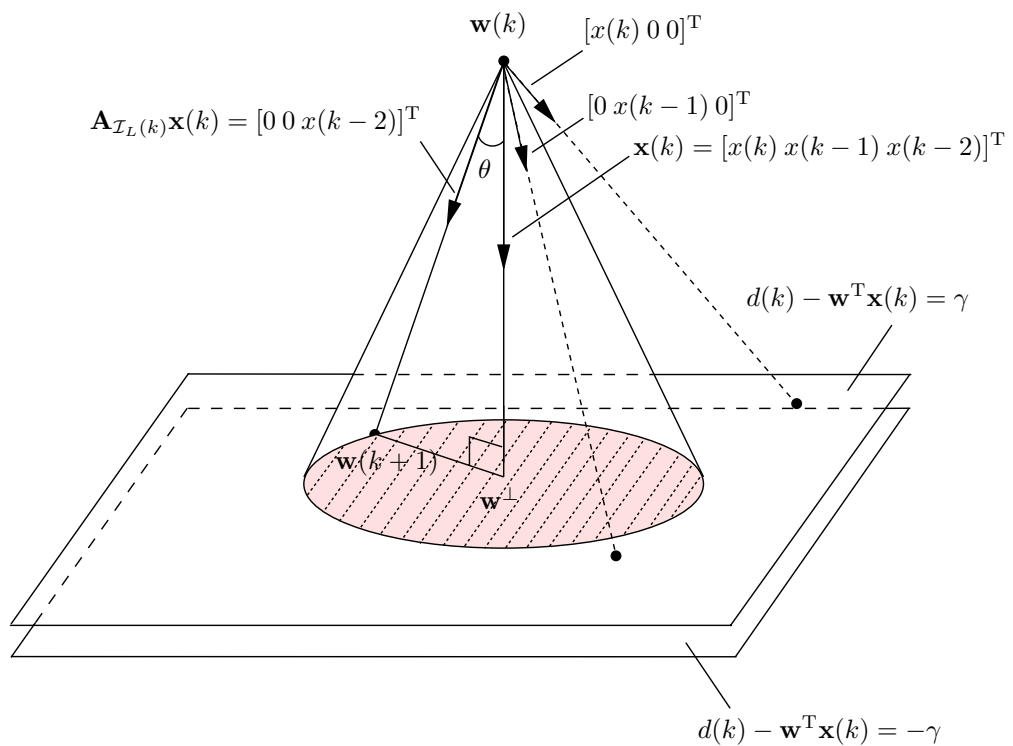


Figure 7.2: Geometric illustration of an update in \mathbb{R}^3 using $L = 1$ coefficient in the partial update, and with $|x(k-2)| > |x(k-1)| > |x(k)|$, the direction of the update is along the vector $[0 \ 0 \ x(k-2)]^T$ forming an angle θ with the input vector $\mathbf{x}(k)$.

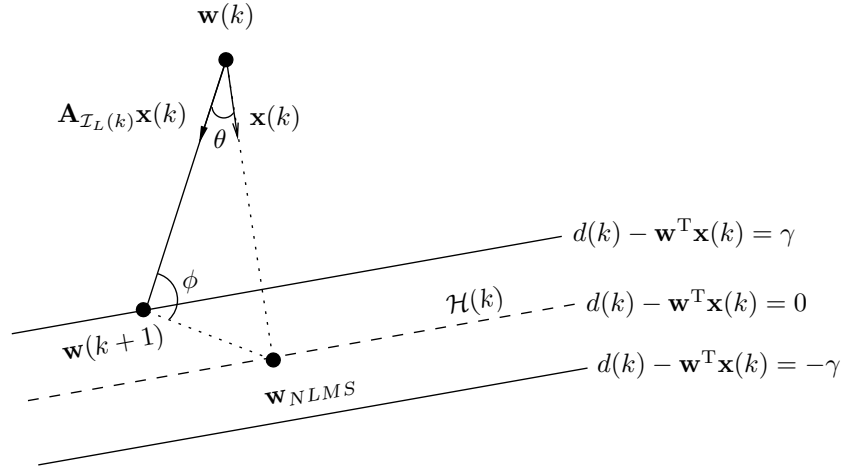


Figure 7.3: General projection solution for $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 \geq \alpha(k)\|\mathbf{x}(k)\|^2$.

update will be closer to the orthogonal solution than the current solution. For large values of N and L , and white input signals, we can make the approximations $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 = L\sigma_x^2$ and $\|\mathbf{x}(k)\|^2 = N\sigma_x^2$, although the former should be considered a rather crude approximation (see Appendix A7.3 of this chapter). Using these approximations, a lower bound on the number of coefficients in the partial update is $L > N/2$. However, it is desirable to allow any choice of smaller values of L , not bounded from below, as long as it does not compromise stability or convergence.

Unlike the PU-NLMS algorithm, the solution to the SM-PU-NLMS algorithm is required to belong to the constraint set. However, stability problems may arise when L is small, and as a consequence, angle θ is increased. In order to address this problem, consider the following update strategies.

Proposition 1. *Increase the number of filter coefficients to update in the partial update vector until the relation $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 \geq \alpha(k)\|\mathbf{x}(k)\|^2$ is true.*

Proposition 1 gives a solution where the number of coefficients in the update vary with time. In case of equality we have $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 = \alpha(k)\|\mathbf{x}(k)\|^2$, and the update can be viewed as the projection of the zero *a posteriori* solution onto $\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)$, as illustrated in Figure 7.3. No upper bound on L is guaranteed, and the proposed strategy would most likely result in L being close to N during the initial adaptation. This is clearly not desirable for the case of partial-update algorithms, where in many cases $L \ll N$ is

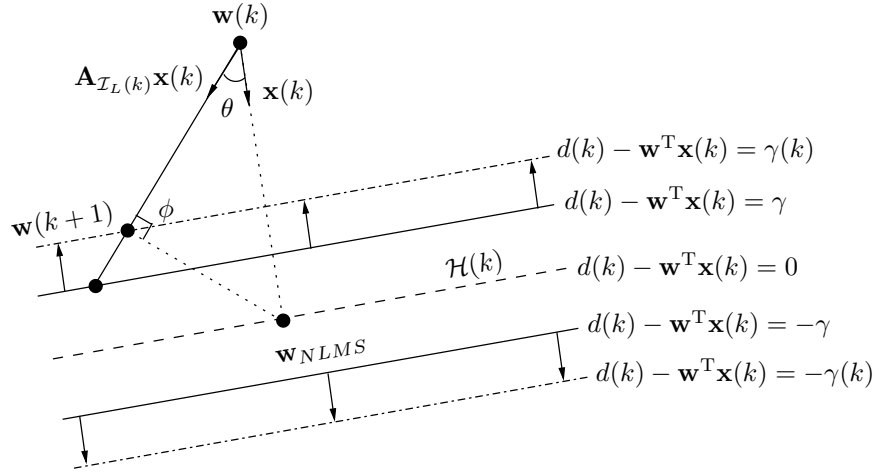


Figure 7.4: Projection solution with temporary expansion of the constraint set $\mathcal{H}(k)$ using a new threshold $\gamma(k)$.

required. Therefore, we consider the following alternative proposition.

Proposition 2. Increase the number of filter coefficients to update in the partial update vector until the relation $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 \geq \alpha(k)\|\mathbf{x}(k)\|^2$ is true or $L = L_{max}$. If $L = L_{max}$, increase the threshold γ temporarily at the k th iteration to $\gamma(k) = \frac{(\|\mathbf{x}(k)\|^2 - \|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2)}{\|\mathbf{x}(k)\|^2}|e(k)|$.

As illustrated in Figure 7.4, Proposition 2 will temporarily expand the constraint set in order to provide a feasible solution if the required number of coefficients to meet Proposition 1 exceeds a predefined maximum number of coefficients L_{max} , set at the design stage. Tables 7.1 and 7.2 show two different versions of the SM-PU-NLMS algorithm. The version in Table 7.1 implements Proposition 2 and the number of coefficients are allowed to vary freely such that $L \leq L_{max}$, where $L_{max} \leq N$ is a predefined value. If $L_{max} = N$ the algorithm will be the same as the one in Proposition 1. Table 7.2 implements a version where L is fixed during the adaptation. The choice between the two versions is application dependent.

7.3.2 Convergence Issues

Any partial update strategy imparts deviation in the direction of update from the direction orthogonal to any hyperplane defined by $d(k) - \mathbf{w}^T\mathbf{x}(k) = c$, where c is a scalar. The angle

of deviation depends on the particular strategy adopted to choose the index set $\mathcal{I}_L(k)$. The strategies suggested in Propositions 1 and 2 above puts a bound on the norm of the update by stating that $\frac{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}{\|\mathbf{x}(k)\|^2} \geq \alpha(k) = 1 - \gamma/|e(k)|$. If this implies violation of condition $L \leq L_{max}$, then the value of γ is temporarily increased (see Figure 7.4). Notice that it may happen that θ approaches 90° , but in these cases the distance to the constraint set boundary will approach zero due to the temporarily increased γ . This is explained by the fact that the angle ϕ in Figure 7.3 is forced to be always greater than or equal to 90° as a consequence of $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2 \geq \alpha(k)$.

A simple model for the desired signal will be adopted for a preliminary study of the convergence properties of the algorithm. For this particular case, let the coefficient-error vector at instant k be defined as $\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$ and the desired signal be modeled as $d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt}$. The error signal is expressed as $e(k) = -\mathbf{x}^T(k)\Delta\mathbf{w}(k)$, and the following expression gives the norm of the coefficient-error vector:

$$\begin{aligned} \|\Delta\mathbf{w}(k+1)\|^2 &= \|\Delta\mathbf{w}(k)\|^2 - \frac{1}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \times \\ &\quad \Delta\mathbf{w}^T(k) \left[\alpha(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\mathbf{x}^T(k) + \alpha(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{A}_{\mathcal{I}_L(k)} - \alpha^2(k)\mathbf{x}(k)\mathbf{x}^T(k) \right] \Delta\mathbf{w}(k) \\ &= \|\Delta\mathbf{w}(k)\|^2 - \frac{1}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \Delta\mathbf{w}^T(k) \left[2\alpha(k)\mathbf{A}_{\mathcal{I}_L(k)} - \alpha^2(k)\mathbf{I} \right] \mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k) \end{aligned} \quad (7.10)$$

A reduction in the coefficient-error norm will occur whenever the term $\Delta\mathbf{w}^T(k) \left[2\alpha(k)\mathbf{A}_{\mathcal{I}_L(k)} - \alpha^2(k)\mathbf{I} \right] \mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)$ is positive. Although matrix $\left[2\alpha(k)\mathbf{A}_{\mathcal{I}_L(k)} - \alpha^2(k)\mathbf{I} \right] \mathbf{x}(k)\mathbf{x}^T(k)$ has nonnegative eigenvalues (see Lemma 4 below) there exist time instants when the coefficient-error norm may increase as a result from the partial-update strategy, as indicated in Figure 7.5. Whenever a reduction in the coefficient-error norm occurs, the optimal $\alpha(k)$ (which causes the largest reduction) is given by the following lemma:

Lemma 4. *The SM-PU-NLMS algorithm with $\alpha(k) = \|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$ achieves the largest reduction in coefficient-error norm whenever a reduction occurs.*

Proof. Matrix $\mathbf{B} = \left[2\alpha(k)\mathbf{A}_{\mathcal{I}_L(k)} - \alpha^2(k)\mathbf{I} \right] \mathbf{x}(k)\mathbf{x}^T(k)$ is a rank-one matrix with the nonzero eigenvalue given by $\lambda = 2\alpha(k)\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2 - \alpha^2(k)\|\mathbf{x}(k)\|^2$. Consequently for

$\alpha(k) \leq 2 \|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$ the relation $\lambda \geq 0$ holds. Maximizing the eigenvalue of matrix \mathbf{B} with respect to $\alpha(k)$ gives $\alpha_{\lambda_{max}}(k) = \|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2/\|\mathbf{x}(k)\|^2$. ■

Notice that, although we cannot guarantee convergence with probability one (see Figure 7.5), we can guarantee almost sure convergence with the heuristic argument that the update, even if only for a fraction of the coefficients, will point toward the optimal solution most of the time. In addition, we can guarantee convergence in the mean-squared sense for the case of additive measurement noise, as stated by the following theorem.

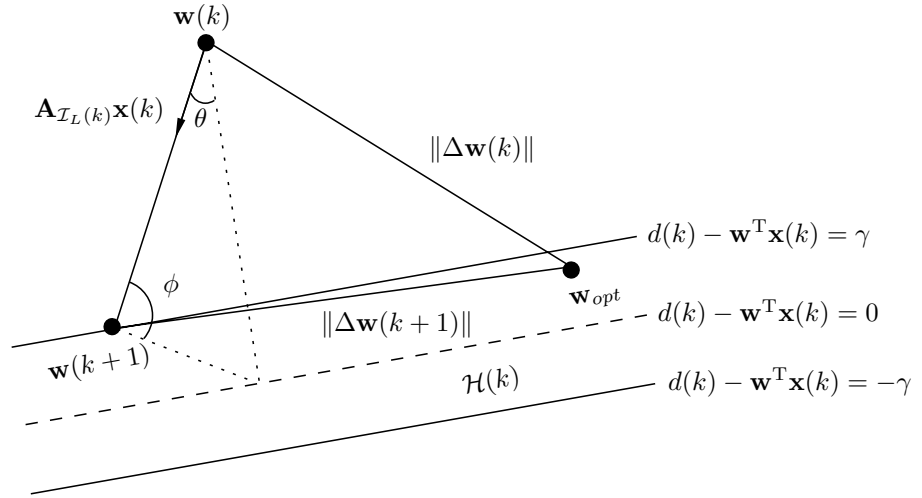


Figure 7.5: Coefficient-error norm evolution.

Theorem 3. *The SM-PU-NLMS converges in the mean-squared sense for zero-mean i.i.d. input signals in the presence of zero-mean additive uncorrelated noise when*

$$\frac{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}{\|\mathbf{x}(k)\|^2} \geq \alpha(k).$$

Proof. In order to account for the data-selectivity of SMF adaptive filters, assign a probability of update $P_e(k) = P(|e(k)| > \gamma)$ and proceed similarly to the derivation described in Appendix A7.2 of this chapter to calculate the coefficient-error norm for the SM-PU-NLMS algorithm:

$$\Delta\mathbf{w}(k+1) = \left[\mathbf{I} - P_e(k)\alpha(k) \frac{\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \right] \Delta\mathbf{w}(k) + P_e(k)\alpha(k) \frac{n(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}. \quad (7.11)$$

With the independence assumption and also assuming the additive measurement noise to be zero mean and not correlated with the white input signal, the expression for the excess MSE is

$$\begin{aligned}
\Delta\xi(k+1) &= \Delta\xi(k) \\
&\quad - \sigma_x^2 \mathbb{E} \left[\frac{\alpha(k) P_e(k) \Delta \mathbf{w}^T(k) \{ \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)} + \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \} \Delta \mathbf{w}(k)}{\| \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \|^2} \right. \\
&\quad \left. - \frac{\alpha^2(k) P_e^2(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\| \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \|^2} \right] + \mathbb{E} \left[\frac{\alpha^2(k) P_e^2(k) n^2(k)}{\| \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \|^2} \right] \\
&= \rho_1 - \rho_2 + \rho_3
\end{aligned} \tag{7.12}$$

The scalar ρ_2 is a linear function of $\Delta\xi(k)$, therefore convergence is ensured if ρ_2 assumes positive values only. Substituting $\alpha(k)$ by its upper bound will account for a worst-case scenario. Invoking the independence assumption and assuming N large such that $\| \mathbf{x}(k) \|^2$ can be considered a reasonable estimate of $(N+1) \mathbb{E} [x^2(k)]$, we may rewrite ρ_2 as

$$\begin{aligned}
\rho_2 &\approx \sigma_x^2 \mathbb{E} \left[\frac{\Delta \mathbf{w}^T(k) P_e(k) (\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)} + \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k)) \Delta \mathbf{w}(k)}{\| \mathbf{x}(k) \|^2} \right. \\
&\quad \left. - \frac{P_e^2(k) \| \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \|^2 \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\| \mathbf{x}(k) \|^4} \right] \\
&\approx \sigma_x^2 \mathbb{E} \left[\frac{\Delta \mathbf{w}^T(k) P_e(k) \mathbb{E} [\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)} + \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k)] \Delta \mathbf{w}(k)}{N \sigma_x^2} \right. \\
&\quad \left. - \frac{P_e^2(k) \mathbb{E} [\| \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \|^2] \Delta \mathbf{w}^T(k) \Delta \mathbf{w}(k)}{N^2 \sigma_x^2} \right]
\end{aligned} \tag{7.13}$$

Evaluating ρ_2 requires the computation of the elements of matrix $\mathbf{B} = \mathbb{E} [\mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)} + \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k)]$. Assuming the input samples to be i.i.d., the off-diagonals will average to zero. Since $\mathbf{A}_{\mathcal{I}_L(k)}$ will select only the L values in the input vector with the largest norm, the diagonal will be an average over the L strongest components only. Let p_i denote the probability for one of the L largest components that contribute to the i th element in the diagonal. Let also $\{y_i\}_{i=1}^N$ be the elements of the input vector $\mathbf{x}(k)$ sorted in magnitude such that $y_1 \leq y_2 \leq \dots \leq y_N$. For a given L , the

diagonal elements of \mathbf{B} can be calculated as follows

$$\mathbb{E} [\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{A}_{\mathcal{I}_L(k)} + \mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)]_{i,i} = 2 \sum_{i=0}^{L-1} \mathbb{E} [p_i y_{N-i}^2] = \frac{2}{N} \mathbb{E} [\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2] \quad (7.14)$$

where for i.i.d. signals $p_i = 1/N$. Substituting this into (7.13) results in

$$\rho_2 \approx P_e(k) [2 - P_e(k)] \frac{\mathbb{E} [\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2]}{N^2\sigma_x^2} \Delta\xi(k) < 2\Delta\xi(k) \quad (7.15)$$

Therefore, Equation (7.12) is always stable. ■

7.4 Computational Complexity

The computational complexities of the PU-NLMS and the SM-PU-NLMS algorithms depend on the number of coefficients to be updated and the search technique for finding the L elements of $\mathbf{x}(k)$ with largest norm. The computational complexities per update in terms of the number of additions, multiplications, and divisions for the NLMS, SM-NLMS, PU-NLMS, and SM-PU-NLMS (L fixed) algorithms are shown in Table 7.3. Although the PU-NLMS and SM-PU-NLMS algorithms have a similar complexity per update, the gain of applying the SM-PU-NLMS algorithm comes through the reduced number of required updates, which cannot be accounted for *a priori*. For time instants where no updates are required, the complexity of the SM-PU-NLMS algorithm is due to filtering, i.e., $N - 1$ additions and N multiplications. In the operation counts, the value of $\|\mathbf{x}(k-1)\|^2$ was assumed known at iteration k such that $\|\mathbf{x}(k)\|^2$ can be computed as $\|\mathbf{x}(k)\|^2 = \|\mathbf{x}(k-1)\|^2 + x^2(k) - x^2(k-N)$, which requires only two multiplications and two additions. In order to find the L largest-norm elements in $\mathbf{x}(k)$, *comparison-sort* algorithms can be used, which require a maximum number comparisons of order $O(N \log N)$. Examples of such comparison-sort algorithms are the Heapsort and the Mergesort algorithms [120, 121]. For the SM-PU-NLMS algorithm, it is necessary that the comparison-sort algorithm run irrespectively if an update is required or not. Both the PU-NLMS and the SM-PU-NLMS algorithms require additional memory to store the pointers to the sorted list. The amount of additional memory required and the number of elements to sort

can be reduced by partitioning the coefficient and input vectors into blocks and perform block-updates as proposed in [118], but at the expense of a decrease in convergence speed.

7.5 Simulation Results

7.5.1 Verification of the Analysis of the PU-NLMS Algorithm

In this subsection, our analysis of the PU-NLMS algorithm is validated using a system-identification setup. The number of coefficients was $N = 51$, and the input signal was zero-mean Gaussian noise with $\sigma_x^2 = 1$. The signal-to-noise ratio (SNR) was set to 60 dB.

Figure 7.6 shows the learning curves for the case of $L = 5$, $L = 10$, and $L = 25$ coefficients in the partial update. The curves were obtained through averaging 100 trials. The step size for each value of L was chosen such that convergence to the same level of misadjustment was achieved. The corresponding theoretical learning curves obtained from evaluating Equation (7.32) were also plotted. As can be seen from the figure, the theoretical curves are very close to the simulations. Figure 7.7 shows the excess MSE as a function of μ ranging from $0.1\mu_{max}$ to $0.8\mu_{max}$ for different values of L , where μ_{max} is given by Equation (7.34) in Appendix A7.2 of this chapter. Note that the axis is normalized with respect to the maximum step size μ_{max} , which is different for each value L . The quantity $E[\tilde{r}^2(k)]$ needed for the calculation of μ_{max} was obtained through numerical integration. For $L = 5$, $L = 10$, and $L = 25$ the corresponding values were $E[\tilde{r}^2(k)] = 21.438$, $E[\tilde{r}^2(k)] = 32.232$, and $E[\tilde{r}^2(k)] = 43.860$, respectively. As can be seen from Figure 7.7, the theoretical results are very close to the simulations within the range of step sizes considered. Using step sizes larger than $0.8\mu_{max}$, resulted in poor accuracy or caused divergence. This is expected due to the approximations made in the analysis. However, only step sizes in the range $\mu \leq 0.5\mu_{max}$ are of practical interest because larger ones will neither increase the convergence speed nor decrease the misadjustment. This fact is illustrated in Figure 7.8, where the theoretical convergence curves were plotted for different values of μ using $L = 10$ and $N = 51$. Therefore, we may state that our theoretical analysis is able to predict very accurately the excess MSE for the whole range of practical step sizes.

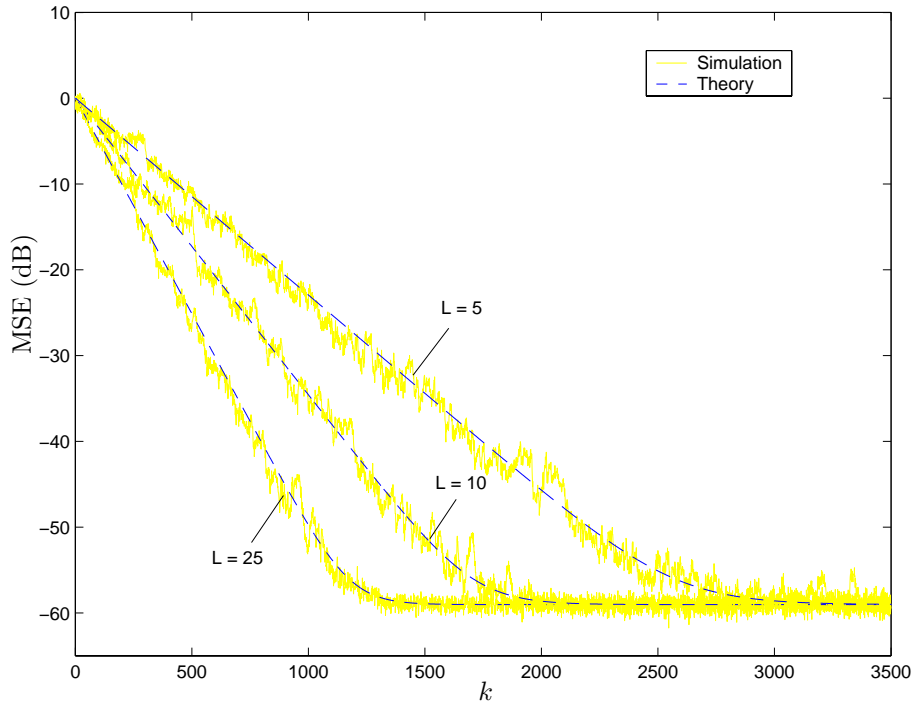


Figure 7.6: Learning curves for the PU-NLMS algorithm for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, $SNR = 60$ dB.

In Figure 7.9 we compare our results (solid lines) with those provided by [14] (dashed lines) for the particular case where their algorithm is equal to the one presented in Section II of this chapter. As seen from Figure 7.9, the results presented in [14] are not accurate even for reasonably high values of L , whereas Figure 7.7 shows that our analysis is accurate for a large range of L . This comes from the fact that in [14] order statistics was not applied in the analysis, resulting in poor estimates of $E[\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2]$ for most values of $L < N$.

7.5.2 SM-PU-NLMS Algorithm

In this section, the two SM-PU-NLMS algorithms are applied to a system identification problem. The number of coefficients was $N = 51$ and colored noise input signal was used with SNR set to 60dB. The colored noise was generated by passing a white noise sequence through a one-pole filter with pole at $z_p = 0.8238$. The bound on the output error was set to $\gamma = \sqrt{5\sigma_n^2}$. Figure 7.10 shows the learning curves averaged over 500 simulations for the SM-PU-NLMS algorithm using the algorithm shown in Table 2, i.e., with L constant.

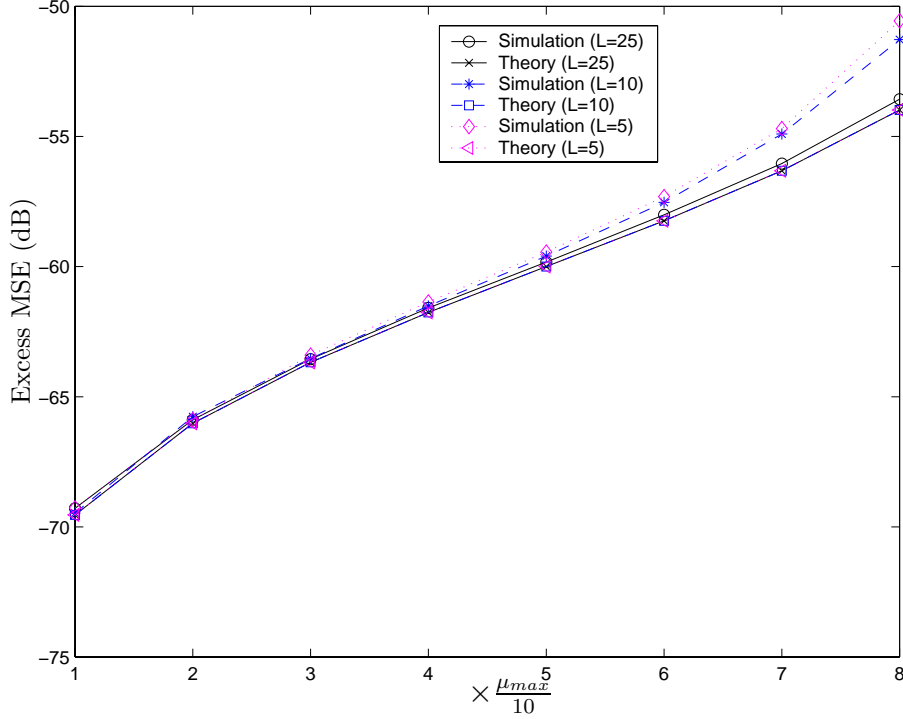


Figure 7.7: Excess MSE for the PU-NLMS algorithm versus the step size μ for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, $SNR = 60$ dB.

The learning curve for the PU-NLMS algorithm with $L = 5$ was included as a reference. The step size in the PU-NLMS algorithm was $\mu = 0.3676$ which resulted in the same level of misadjustment as the SM-PU-NLMS algorithm with $L = 5$. In 12000 iterations the number of times that an update took place for $L = 5$, $L = 10$ and $L = 25$ were 4950, 3340, and 2420, respectively. This should be compared to the 12000 updates required by the PU-NLMS algorithm, and the drastic reduction is a result of the SMF strategy. As can be seen from Figure 7.10 the SM-PU-NLMS algorithm converges faster than the PU-NLMS algorithm for the same level of misadjustment with less computational complexity.

Figure 7.11 shows the learning curves for the SM-PU-NLMS algorithm with variable L . The counterpart results for the SM-PU-NLMS algorithm obtained previously are included in Figure 7.11 for reference. As can be seen from the figure, the SM-PU-NLMS algorithm with variable L converges to a slightly higher steady-state value than the SM-PU-NLMS algorithm using fixed L . In 12000 iterations the number of times that an update took place for $L_{max} = 5$, $L_{max} = 10$, and $L_{max} = 25$ were 5070 and 3640, and 2840, respectively,

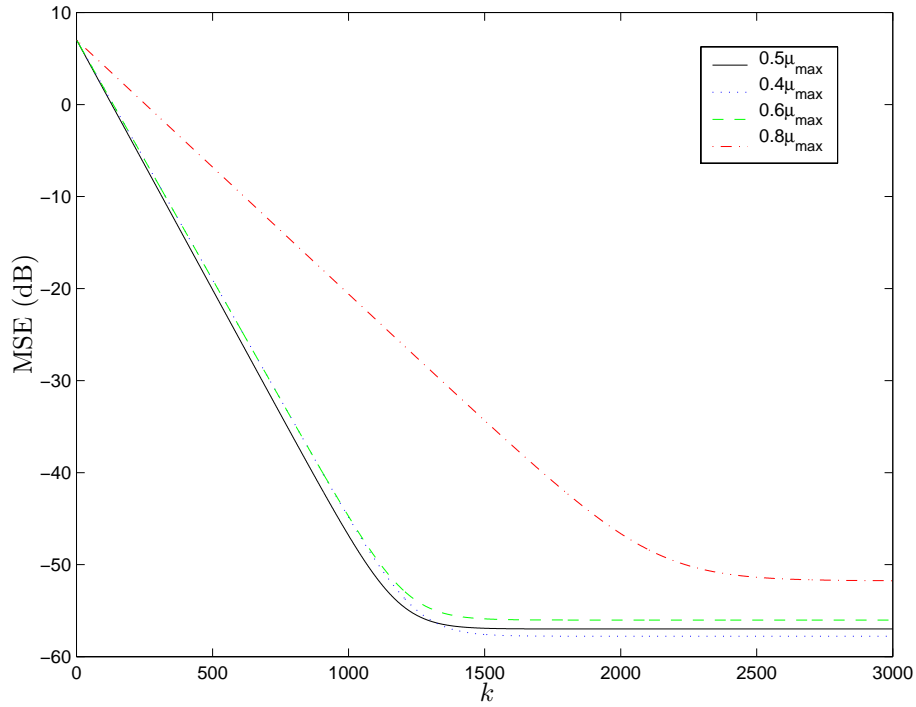


Figure 7.8: Theoretical learning curves for different choice of step size in the PU-NLMS algorithm for $N = 51$ and $L = 10$, $SNR = 60$ dB.

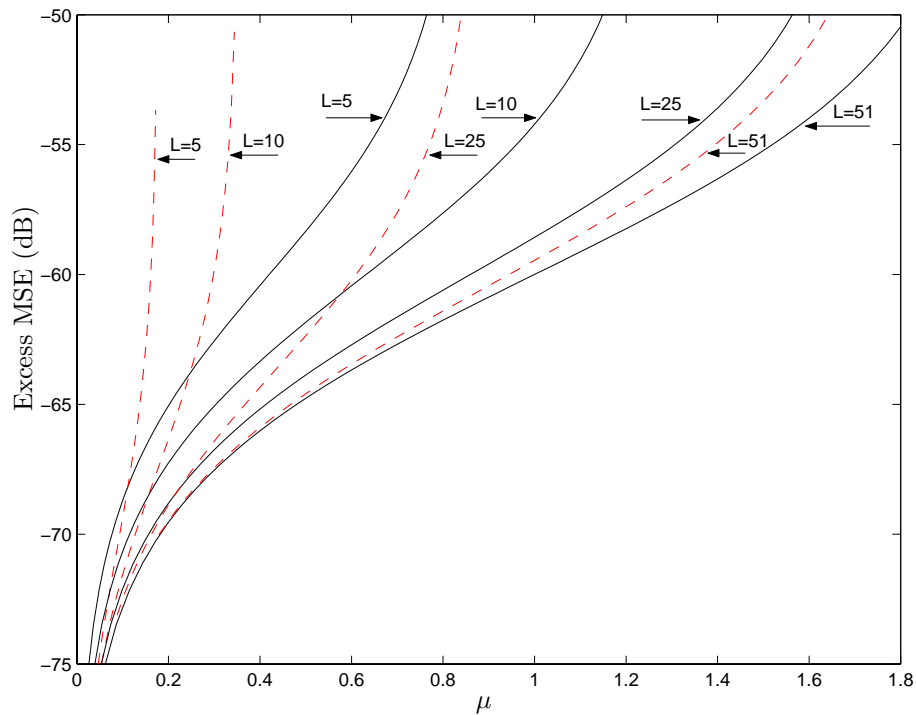


Figure 7.9: Comparison of Equation (7.32) (solid lines) with the excess MSE formula obtained from literature (dashed lines).

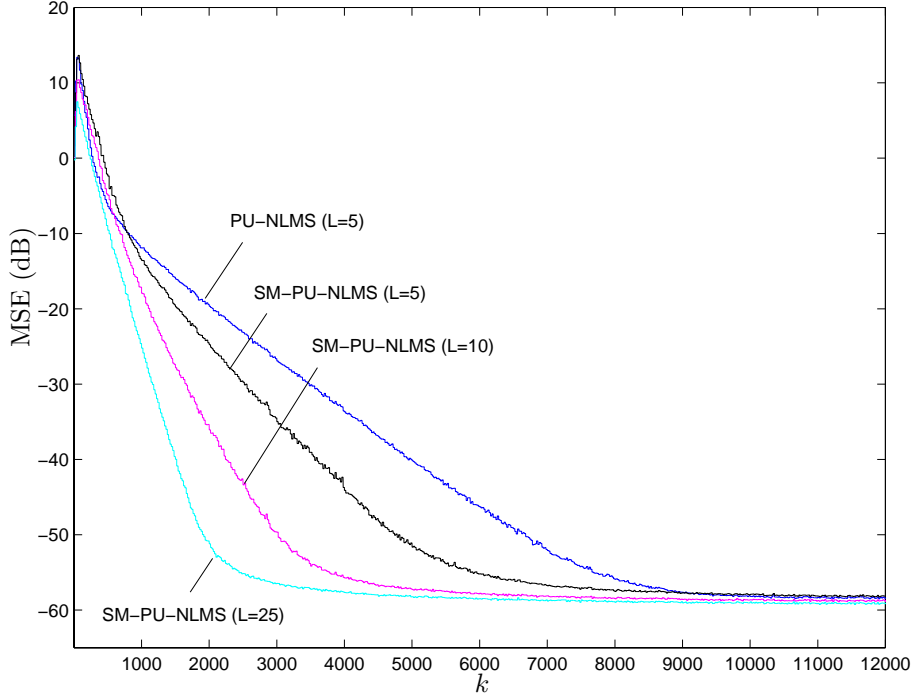


Figure 7.10: Learning curves for the SM-PU-NLMS algorithm using fixed L for $N = 51$, $L = 5$, $L = 10$ and $L = 25$, and the PU-NLMS algorithm for $L = 5$, $SNR = 60$ dB.

which is slightly higher than when L is fixed. However, the number of coefficients in the partial update was also smaller for most of the time instants, which can be observed from Figure 7.12 where for $L \leq L_{max} = 25$, the number of coefficients in the partial update versus time is shown during one realization. As can be seen from the figure, a number of coefficients close to $L_{max} = 25$ coefficients were updated during the initial iterations whereas later on this value decreases. The same trend was observed for the case of $L_{max} = 5$ and $L_{max} = 10$.

7.6 Conclusions

This chapter studied normalized partial-update adaptation algorithms. Convergence analysis for the conventional partial-update NLMS (PU-NLMS) algorithm was presented, which gave further insight to the algorithm in terms of stability, transient and steady-state performances. The analysis was validated through simulations showing excellent agreement. New stability bounds were given for the step size that controls the stability, convergence

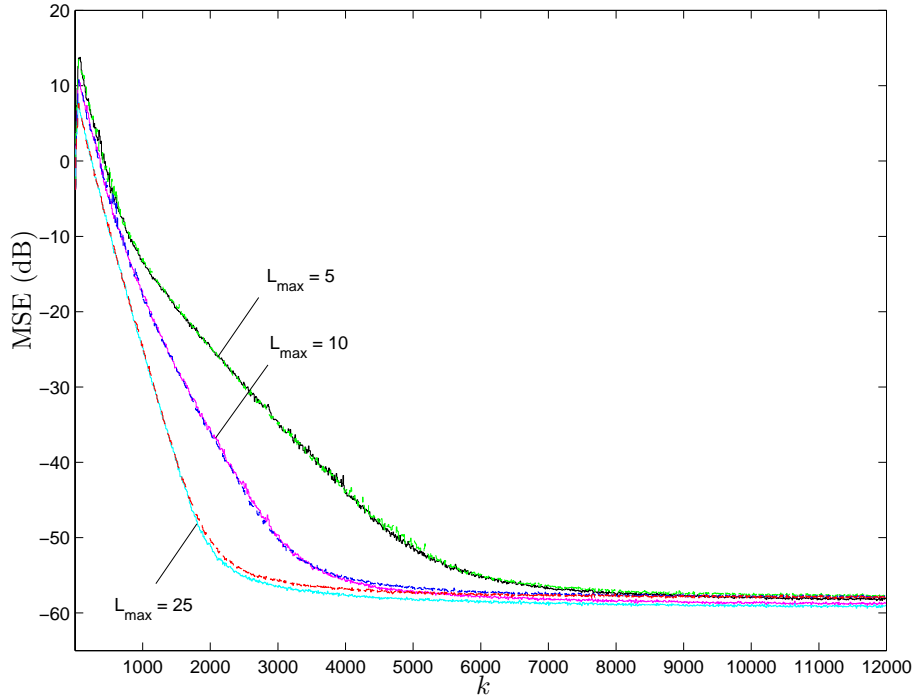


Figure 7.11: Learning curves, $L \leq L_{max}$ (dashed) and L fixed (solid).

speed, and final excess MSE of the PU-NLMS algorithm. It was shown that the step size giving the fastest convergence could be related to the time-varying step size of the M-Max NLMS algorithm. These results extend and improve the accuracy of the previous results reported in the literature. The excellent agreement between the theory and the simulations presented here for the PU-NLMS algorithm has advanced significantly the study of order-statistic-based adaptive filtering algorithms. Furthermore, novel data-selective normalized adaptation algorithms with partial updating were derived based on the concept of set-membership filtering.

The new algorithms benefit from the reduced average computational complexity from the set-membership filtering framework and the reduced computational complexity resulting from partial updating. Simulations were presented for a system identification application. It was verified that not only the data-selective adaptation algorithms with partial updating can further reduce the computational complexity when compared to the partial-update NLMS algorithm, but can also present a faster convergence for the same level of excess MSE. The new SM-PU-NLMS algorithm proposed and discussed herein shows a new perspective for adaptive filters when a very large number of coefficients needs to be used

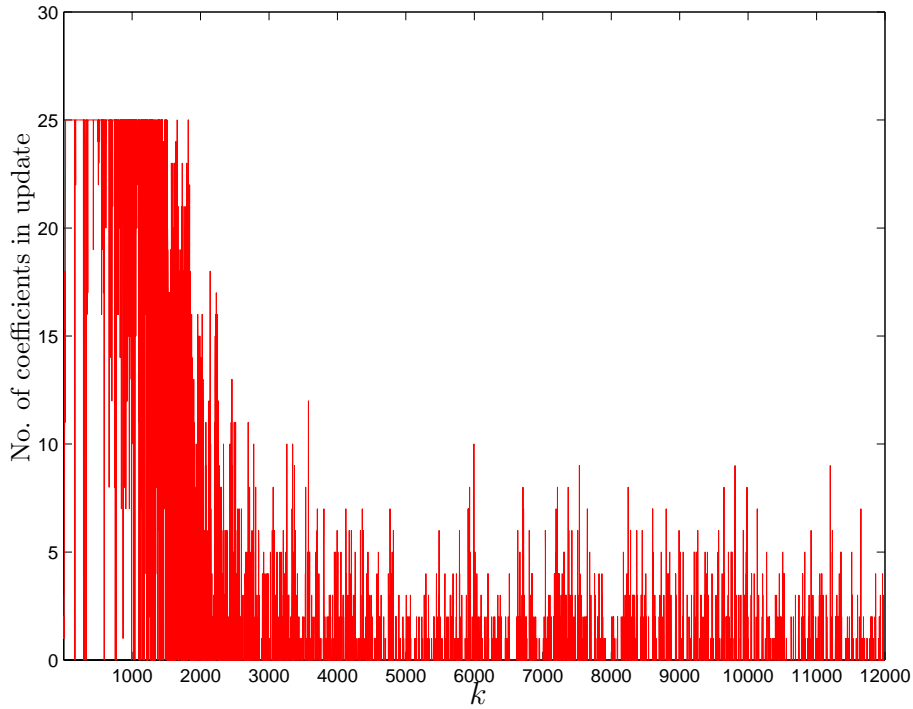


Figure 7.12: Number of coefficients updated in the partial update versus time in a single realization for the SM-PU-NLMS algorithm with L variable and $L \leq 25$.

and high performance needs to be maintained.

Appendix A7.1

The optimization problem in (7.1) can be solved by the method of Lagrange multipliers having the following objective function

$$J_{\mathbf{w}} = \|\mathbf{w} - \mathbf{w}(k)\|^2 + \lambda_1 [d(k) - \mathbf{x}^T(k)\mathbf{w}] + \boldsymbol{\lambda}_2^T \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} [\mathbf{w} - \mathbf{w}(k)]. \quad (7.16)$$

where λ_1 is a scalar and $\boldsymbol{\lambda}_2$ is an $N \times 1$ vector. Setting the derivative of (7.16) with respect to \mathbf{w} equal to zero and solving for the new coefficient vector gives us

$$\mathbf{w} = \mathbf{w}(k) + \frac{\lambda_1}{2} \mathbf{x}(k) - \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \frac{\boldsymbol{\lambda}_2}{2} \quad (7.17)$$

In order to solve for the constraints, multiply Equation (7.17) by $\tilde{\mathbf{A}}_{\mathcal{I}_L(k)}$ and subtract $\tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \mathbf{w}(k)$ from both sides, i.e.,

$$\tilde{\mathbf{A}}_{\mathcal{I}_L(k)} [\mathbf{w} - \mathbf{w}(k)] = \mathbf{0} = +\frac{\lambda_1}{2} \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \mathbf{x}(k) - \tilde{\mathbf{A}}_{\mathcal{I}_L(k)} \frac{\boldsymbol{\lambda}_2}{2} \quad (7.18)$$

where we have used $\tilde{\mathbf{A}}_{\mathcal{I}_L(k)}\tilde{\mathbf{A}}_{\mathcal{I}_L(k)} = \tilde{\mathbf{A}}_{\mathcal{I}_L(k)}$. Therefore,

$$\tilde{\mathbf{A}}_{\mathcal{I}_L(k)}\frac{\lambda_2}{2} = \frac{\lambda_1}{2}\tilde{\mathbf{A}}_{\mathcal{I}_L(k)}\mathbf{x}(k) \quad (7.19)$$

If we substitute (7.19) in (7.17) we get

$$\begin{aligned} \mathbf{w} &= \mathbf{w}(k) + \frac{\lambda_1}{2}\mathbf{x}(k) - \frac{\lambda_1}{2}\tilde{\mathbf{A}}_{\mathcal{I}_L(k)}\mathbf{x}(k) \\ &= \mathbf{w}(k) + \mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\frac{\lambda_1}{2} \end{aligned} \quad (7.20)$$

where we used $\mathbf{A}_{\mathcal{I}_L(k)} = \mathbf{I} - \tilde{\mathbf{A}}_{\mathcal{I}_L(k)}$. Finally, λ_1 is obtained by pre-multiplying (7.20) by $\mathbf{x}^T(k)$, which gives $\lambda_1/2 = e(k)/\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2$ for $\mathbf{x}^T(k)\mathbf{w}(k) = d(k)$. Our final update is then given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{e(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}. \quad (7.21)$$

Appendix A7.2

In this appendix, the PU-NLMS algorithm is analyzed in the mean-squared sense.

Coefficient-Error Vector

In order to derive expressions for the second-order statistics of the PU-NLMS algorithm we will first derive an expression for the evolution of the coefficient-error vector. Assuming that the desired signal is given by

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_{opt} + n(k) \quad (7.22)$$

and defining the coefficient error vector as $\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_{opt}$, we can express the error as

$$e(k) = n(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k) \quad (7.23)$$

Therefore, from Equations (7.4) and (7.23) we have

$$\Delta\mathbf{w}(k+1) = \left[\mathbf{I} - \mu \frac{\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2} \right] \Delta\mathbf{w}(k) + \mu \frac{n(k)\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2}. \quad (7.24)$$

Excess MSE for White Input Signals

For the MSE analysis, we assume that the vectors are excited in a discrete number of directions. This model was used to analyze the NLMS algorithm in [44] and is consistent with the first- and second-order statistics of the original input signal. The model was also successfully used to analyze the quasi-Newton (QN) [43] and the binormalized data-reusing LMS (BNDRLMS) [51] algorithms.

The following assumptions are made:

- Independence between $\mathbf{x}(k)$ and $\Delta\mathbf{w}(k)$.
- The vectors $\mathbf{x}(k)$ and $\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)$ are modeled by $\mathbf{x}(k) = s(k)r(k)\mathbf{v}(k)$ and $\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k) = \tilde{s}(k)\tilde{r}(k)\tilde{\mathbf{v}}(k)$, respectively, where
 - $s(k)$ and $\tilde{s}(k)$ take on values ± 1 with probability $1/2$.
 - $r(k)$ and $\tilde{r}(k)$ are positive real valued stochastic variables such that $r^2(k)$ and $\tilde{r}^2(k)$ have the same probability distribution functions as $\|\mathbf{x}(k)\|^2$ and $\|\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\|^2$, respectively.
 - $\mathbf{v}(k)$ is equal to one of the N orthonormal eigenvectors of $\mathbf{R} = \mathbb{E}[\mathbf{x}(k)\mathbf{x}^T(k)]$ denoted as $\mathcal{V}(k)$, and $\tilde{\mathbf{v}}(k)$ is equal to one of the $N+1$ orthonormal eigenvectors of $\tilde{\mathbf{R}} = \mathbb{E}[\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{A}_{\mathcal{I}_L(k)}^T]$ denoted as $\tilde{\mathcal{V}}(k)$. For white Gaussian input signals $\mathbf{v}(k)$ and $\tilde{\mathbf{v}}(k)$ are uniformly distributed and \mathbf{R} and $\tilde{\mathbf{R}}$ share the same eigenvectors, i.e., $\mathcal{V}(k) = \tilde{\mathcal{V}}(k)$. Therefore,

$$P[\mathbf{v}(k) = \mathcal{V}(k)] = P[\tilde{\mathbf{v}}(k) = \mathcal{V}(k)] = \frac{1}{N} \quad (7.25)$$

Notice that for any value of L we have $s(k) = \tilde{s}(k)$ since the inner product of $\mathbf{x}(k)$ and $\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)$ is always positive.

For white input signals, the excess MSE is given by $\Delta\xi(k+1) = \sigma_x^2 \text{tr}[\text{cov}(\Delta\mathbf{w}(k+1))] [1]$,

where

$$\begin{aligned}
\text{cov} [\Delta \mathbf{w}(k+1)] &= \text{E} [\Delta \mathbf{w}(k+1) \Delta \mathbf{w}^T(k+1)] \\
&= \text{E} \left[\left(\mathbf{I} - \mu \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \left(\mathbf{I} - \mu \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right)^T \right] \\
&\quad + \text{E} \left[\mu^2 n^2(k) \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right] \\
&= \text{E} [\Delta \mathbf{w}_k \Delta \mathbf{w}^T(k)] \\
&\quad - \text{E} \left[\mu \frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right] \\
&\quad - \text{E} \left[\mu \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right] \\
&\quad + \text{E} \left[\mu^2 \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right] \\
&\quad + \text{E} \left[\mu^2 n^2(k) \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right]
\end{aligned} \tag{7.26}$$

Let us analyze each term separately:

$$\psi_1 = \sigma_x^2 \text{tr} \left\{ \text{E} [\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k)] \right\} = \Delta \xi(k) \tag{7.27}$$

$$\begin{aligned}
\psi_2 &= \sigma_x^2 \text{tr} \left\{ \text{E} \left[\mu \frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right] \right\} \\
&= \sigma_x^2 \text{tr} \left\{ \text{E} \left[\mu \frac{\Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) s(k) r(k) \tilde{s}(k) \tilde{r}(k) \mathbf{v}(k) \tilde{\mathbf{v}}^T(k)}{\tilde{s}^2(k) \tilde{r}^2(k)} \right] \right\} \\
&= \sigma_x^2 \text{E} \left[\mu \Delta \mathbf{w}^T(k) \mathbf{v}(k) \tilde{\mathbf{v}}^T(k) \Delta \mathbf{w}(k) \right] \text{E} \left[\frac{s(k) r(k) \tilde{s}(k) \tilde{r}(k)}{\tilde{r}^2(k)} \right] \\
&= \frac{\mu}{N} \text{E} \left[\frac{r(k)}{\tilde{r}(k)} \right] \Delta \xi(k)
\end{aligned} \tag{7.28}$$

where we used Equation (7.25). Since $\text{tr} \{\mathbf{A}\mathbf{B}\} = \text{tr} \{\mathbf{B}\mathbf{A}\}$ we will have $\psi_2 = \psi_3$.

$$\begin{aligned}
\psi_4 &= \sigma_x^2 \mu^2 \text{tr} \left\{ \text{E} \left[\frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k) \Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right] \right\} \\
&= \sigma_x^2 \mu^2 \text{E} \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T \mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right]
\end{aligned}$$

$$\begin{aligned}
&= \sigma_x^2 \mu^2 \mathbb{E} \left[\frac{\Delta \mathbf{w}^T(k) \mathbf{x}(k) \mathbf{x}^T(k) \Delta \mathbf{w}(k)}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^2} \right] \\
&= \sigma_x^2 \mu^2 \mathbb{E} \left[\Delta \mathbf{w}^T(k) \mathbf{v}(k) \mathbf{v}^T(k) \Delta \mathbf{w}(k) \frac{r^2(k)}{\tilde{r}^2(k)} \right] \\
&= \mu^2 \frac{1}{N} \mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right] \Delta \xi(k)
\end{aligned} \tag{7.29}$$

$$\begin{aligned}
\psi_5 &= \sigma_x^2 \mu^2 \text{tr} \left\{ \mathbb{E} \left[n^2(k) \frac{\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{A}_{\mathcal{I}_L(k)}^T}{\|\mathbf{A}_{\mathcal{I}_L(k)} \mathbf{x}(k)\|^4} \right] \right\} \\
&= \sigma_x^2 \sigma_n^2 \mu^2 \text{tr} \left\{ \mathbb{E} \left[\tilde{\mathbf{v}}(k) \tilde{\mathbf{v}}^T(k) \frac{1}{\tilde{r}^2(k)} \right] \right\} \\
&= \sigma_x^2 \sigma_n^2 \mu^2 \mathbb{E} \left[\frac{1}{\tilde{r}^2(k)} \right]
\end{aligned} \tag{7.30}$$

Finally we obtain the expression for the excess MSE

$$\begin{aligned}
\Delta \xi(k+1) &\approx \psi_1 - \psi_2 - \psi_3 + \psi_4 + \psi_5 \\
&= \left\{ 1 - \frac{\mu}{N} \left(2 \mathbb{E} \left[\frac{r(k)}{\tilde{r}(k)} \right] - \mu \mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right] \right) \right\} \Delta \xi(k) + \mu^2 \sigma_x^2 \sigma_n^2 \mathbb{E} \left[\frac{1}{\tilde{r}^2(k)} \right]
\end{aligned} \tag{7.31}$$

which can be approximated as

$$\Delta \xi(k+1) \approx \left\{ 1 - \frac{\mu}{N} \left(2 - \mu \mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right] \right) \right\} \Delta \xi(k) + \mu^2 \sigma_x^2 \sigma_n^2 \mathbb{E} \left[\frac{1}{\tilde{r}^2(k)} \right] \tag{7.32}$$

where the conservative approximation $\mathbb{E} \left[\frac{r(k)}{\tilde{r}(k)} \right] \approx 1$ was used. The stability region in the mean-squared sense for μ is

$$0 < \mu < \frac{2}{\mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right]} \tag{7.33}$$

where the step size $\mu = 1/\mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right]$ yields maximum reduction of $\Delta \xi(k)$ in (7.32). Further simplifications with $\mathbb{E} \left[\frac{r^2(k)}{\tilde{r}^2(k)} \right] \approx \frac{\mathbb{E}[r^2(k)]}{\mathbb{E}[\tilde{r}^2(k)]}$ give us

$$0 < \mu < \frac{2 \mathbb{E}[\tilde{r}^2(k)]}{N \sigma_x^2} \tag{7.34}$$

where $\mathbb{E}[r^2(k)] = N \sigma_x^2$ and $\mathbb{E}[\tilde{r}^2(k)]$ can be calculated using knowledge of L and N using order statistics (see also Appendix A7.3 of this chapter). A more pessimistic bound can be obtained by using the relation $\mathbb{E}[\tilde{r}^2(k)] \geq L \sigma_x^2$ (see Appendix A7.3 of this chapter) giving

$$0 < \mu < \frac{2L}{N} \tag{7.35}$$

which corresponds to the bound given in [14]. We stress that the analysis presented in this appendix shows that step sizes larger than the ones indicated by Equation (7.35) may be used according to Equation (7.34).

For $k \rightarrow \infty$ we have

$$\begin{aligned} \Delta\xi_{exc} &\approx N \frac{\mu\sigma_n^2\sigma_x^2}{2 - \mu\mathbb{E}\left[\frac{r^2(k)}{\tilde{r}^2(k)}\right]} \mathbb{E}\left[\frac{1}{\tilde{r}^2(k)}\right] \\ &\approx N \frac{\mu\sigma_n^2\sigma_x^2}{2\mathbb{E}[\tilde{r}^2(k)] - \mu\mathbb{E}[r^2(k)]} \\ &= N \frac{\mu\sigma_n^2\sigma_x^2}{2\mathbb{E}[\tilde{r}^2(k)] - \mu N\sigma_x^2}. \end{aligned} \tag{7.36}$$

Appendix A7.3

In this Appendix it is shown how to obtain numerically $\mathbb{E}[\tilde{r}^2(k)]$ used in the step size bound derived in Appendix A7.2 of this chapter. In addition, a lower bound on $\mathbb{E}[\tilde{r}^2(k)]$ is provided. This parameter was also required in the analysis of the M-Max NLMS algorithm [116], which used the approach as presented here.

The basic problem here is to calculate the second moment of ordered statistics. This problem has received much attention in the past, see, e.g., [122, 123, 124], where recursion formulas and tables were produced for expected values and moments of ordered statistics for various different distributions.

Let $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_N]^T$ be a vector containing the elements of vector $\mathbf{x}(k) = [x(k) \ x(k-1) \ \dots \ x(k-N+1)]^T$ ordered in value, i.e, $y_1 \leq y_2 \leq \dots \leq y_j \leq \dots \leq y_N$. The probability density function $f_j(y)$ of the j th element in \mathbf{y} is given by [125]

$$f_j(y) = \frac{N!}{(j-1)!(N-j)!} F_x^{j-1}(y) [1 - F_x(y)]^{N-j} f_x(y) \tag{7.37}$$

where $f_x(x)$ is the density of the unsorted random variables in vector $\mathbf{x}(k)$ and $F_x^{j-1}(x)$ is their cumulative distribution to the power of $j-1$. The second moment of the j th element is given by

$$\begin{aligned} \mathbb{E}[y_j^2] &= \int_{-\infty}^{\infty} y_j^2 f_j(y) dy \\ &= \frac{N!}{(j-1)!(N-j)!} \int_{-\infty}^{\infty} y^2 F_x^{j-1}(y) [1 - F_x(y)]^{N-j} f_x(y) dy \end{aligned} \tag{7.38}$$

The PU-NLMS algorithm chooses the L elements in $\mathbf{x}(k)$ of largest magnitude. Therefore, if we order the values in $\mathbf{A}_{\mathcal{I}_L(k)}\mathbf{x}(k)$ in magnitude their second moments can be found by evaluating (7.38) for $j = N + 1 - L, \dots, N$. For the case of Gaussian input signals and using the cumulative distribution and density functions for the magnitude of a Gaussian variable, we have

$$F_x(y) = \begin{cases} 2\Phi_x(y) - 1 & \text{for } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.39)$$

and

$$f_x(y) = \begin{cases} 2\phi_x(y) & \text{for } y \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.40)$$

where $\Phi_x(y)$ and $\phi_x(y)$ are the cumulative distribution function and the density function, respectively, of a Gaussian variable. The density function $f_x(y)$ in (7.40) is in fact the probability density function for a random variable from a chi-distribution with one degree of freedom. The problem of calculating moments of order statistics in samples from the chi-distribution (1 degree of freedom) was considered in [124], where a recursion formula was developed. The quantity $E[\tilde{r}^2(k)]$ is given by

$$E[\tilde{r}^2(k)] = \sum_{j=N+1-L}^N \frac{2N!}{(j-1)!(N-j)!} \int_0^\infty y^2 [2\Phi_x(y) - 1]^{j-1} [2 - 2\Phi_x(y)]^{N-j} \phi_x(y) dy \quad (7.41)$$

which for given N and L can be evaluated numerically.

With the aid of the previous results we are able to calculate bounds for $E[\tilde{r}^2(k)]$, as stated in the following lemma.

Lemma 5. *If the input signal $x(k)$ is Gaussian with zero mean and variance σ_x^2 , then $E[\tilde{r}^2(k)] = \sum_{j=N+1-L}^N E[y_j^2]$, where $L \leq N$, is bounded as follows:*

$$L\sigma_x^2 \leq E[\tilde{r}^2(k)] \leq N\sigma_x^2$$

with equality iff $L = N$.

Proof. In the proof we need the following relations

$$E[y_1^2] \leq E[y_2^2] \leq \dots \leq E[y_k^2] \leq \dots \leq E[y_N^2] \quad (7.42)$$

$$\sum_{k=1}^N \mathbb{E}[y_k^2] = N\mathbb{E}[x^2(k)] \quad (7.43)$$

$$\mathbb{E}[y_1^2] < \sigma_x^2, \text{ for } N > 1 \quad (7.44)$$

Relation (7.42) holds true by definition, and (7.43) holds true for an arbitrary distribution for which the integral in (7.38) converges [124], as shown below

$$\begin{aligned} \sum_{j=1}^N \mathbb{E}[y_j^2] &= \sum_{j=1}^N \frac{N!}{(j-1)!(N-j)!} \int_{-\infty}^{\infty} y^2 F_x^{j-1}(y) [1 - F_x(y)]^{N-j} f_x(y) dy \\ &= \int_{-\infty}^{\infty} \left\{ \sum_{j=1}^N \frac{N!}{(j-1)!(N-j)!} F_x^{j-1}(y) [1 - F_x(y)]^{N-j} \right\} y^2 f_x(y) dy \\ &= \int_{-\infty}^{\infty} \left\{ N \sum_{k=0}^{N-1} \frac{(N-1)!}{k!(N-1-k)!} F_x^k(y) [1 - F_x(y)]^{N-1-k} \right\} y^2 f_x(y) dy \\ &= \int_{-\infty}^{\infty} N y^2 f_x(y) dy \\ &= N\mathbb{E}[x^2(k)] \end{aligned} \quad (7.45)$$

where we used $\sum_{k=0}^N \binom{N}{k} p^k q^{N-k} = (p+q)^N$. The relation (7.44) can be shown for Gaussian input signals by evaluating (7.38) for $j = 1$ with $F_x(y)$ and $f_x(y)$ given by Equations (7.39) and (7.40):

$$\begin{aligned} \mathbb{E}[y_1^2] &= \int_0^{\infty} 2N y^2 [2 - 2\Phi_x(y)]^{N-1} \phi_x(y) dy \\ &= \int_0^{\infty} 2^N N y^2 [1 - \Phi_x(y)]^{N-1} \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{y^2}{2\sigma_x^2}} dy \\ &\leq \int_0^{\infty} 2^N N y^2 \left[\frac{1}{2} e^{-\frac{y^2}{2\sigma_x^2}} \right]^{N-1} \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{y^2}{2\sigma_x^2}} dy \\ &= \int_0^{\infty} 2N y^2 \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{Ny^2}{2\sigma_x^2}} dy \\ &= \frac{\sigma_x^2}{\sqrt{N}} < \sigma_x^2 \text{ for } N > 1. \end{aligned} \quad (7.46)$$

where we used $1 - \Phi_x(y) \leq \frac{1}{2} e^{-\frac{y^2}{2\sigma_x^2}}$ for $y \geq 0$ [125], and $\int_0^{\infty} y^2 e^{-ay^2} dy = \frac{1}{4a} \sqrt{\frac{\pi}{a}}$. From relations (7.42) and (7.44) it follows that $L\sigma_x^2 \leq \mathbb{E}[\tilde{r}^2(k)] \leq N\sigma_x^2$ holds true for $L \leq N$. Relation (7.44) gives us strict inequality for $L < N$ when $N > 0$, and consequently equality holds true only when $L = N$. ■

Table 7.1: SM-PU-NLMS Algorithm, L Time-Varying with $L \leq L_{max}$.

| SM-PU-NLMS Algorithm $L \leq L_{max}$ |
|---|
| <pre> for each k { $P_x = P_x + x^2(k) - x^2(k - N)$ $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { $[\mathbf{z}, \mathbf{i}] = \text{sort} [\mathbf{x}(k)]$ % \mathbf{z}, \mathbf{i}: sorted vector and index vector $l = 0$ % no. of coefficients in PU $p_x = 0$ % power in PU $\alpha(k) = 1 - \gamma/ e(k)$ $b = \alpha(k) \cdot P_x$ while $p_x < b$ and $l < L_{max}$ % too low power? { $l = l + 1$ % increase no. of coefficients in PU $p_x = p_x + z^2(l)$ % power in PU } if $p_x < b$ % increase bound? { $\gamma(k) = [1 - p_x/P_x] \cdot e(k)$ $\alpha(k) = 1 - \gamma(k)/ e(k)$ % equal to p_x/P_x } % update coefficients specified by \mathbf{i} $\mathbf{w}[\mathbf{i}(1 : l)] = \mathbf{w}[\mathbf{i}(1 : l)] + \alpha(k)e(k)/p_x\mathbf{x}[\mathbf{i}(1 : l)]$ } else { $\mathbf{w}(k + 1) = \mathbf{w}(k)$ } } </pre> |

Table 7.2: SM-PU-NLMS Algorithm, L Fixed During the Adaptation.

| SM-PU-NLMS Algorithm L fixed |
|--|
| <pre> for each k { $P_x = P_x + x^2(k) - x^2(k - N)$ $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$ if $e(k) > \gamma$ { [\mathbf{z}, \mathbf{i}] = sort [$\mathbf{x}(k)$] % \mathbf{z}, \mathbf{i}: sorted vector and index vector $p_x = \mathbf{z}^T(1:L)\mathbf{z}(1:L)$ % power in PU $\alpha(k) = 1 - \gamma/ e(k)$ $b = \alpha(k) \cdot P_x$ if $p_x < b$ % increase bound? { $\gamma(k) = [1 - p_x/P_x] \cdot e(k)$ $\alpha(k) = 1 - \gamma(k)/ e(k)$ % equal to p_x/P_x } % update coefficients specified by \mathbf{i} $\mathbf{w}[\mathbf{i}(1:l)] = \mathbf{w}[\mathbf{i}(1:l)] + \alpha(k)e(k)/p_x\mathbf{x}[\mathbf{i}(1:l)]$ } else { $\mathbf{w}(k+1) = \mathbf{w}(k)$ } } </pre> |

Table 7.3: Computational Complexity.

| ALG. | MULT. | ADD. | DIV. |
|--------------|-------------|-------------|------|
| NLMS | $2N + 3$ | $2N + 2$ | 1 |
| SM-NLMS | $2N + 3$ | $2N + 3$ | 2 |
| PU-NLMS [14] | $N + L + 3$ | $N + L + 2$ | 1 |
| SM-PU-NLMS | $N + L + 3$ | $N + L + 3$ | 2 |

Chapter 8

Conclusions and Future Work

This chapter concludes the results of the thesis and suggests a number of future directions that can be taken.

8.1 Conclusions

The thesis considered the derivation of new adaptive filtering algorithms for two types of adaptive filters: the linearly-constrained adaptive filter and the conventional training-based adaptive filter. When choosing an adaptive filtering algorithm, it is important to take into account its computational complexity, convergence speed and misadjustment. These performance measures have a direct effect on the applicability of the particular algorithm to the application in mind. For example, in communications systems the amount of training-data may be limited to a few samples. Therefore, a fast convergence speed of the adaptive filtering algorithm can be crucial. Low computational complexity of the adaptive filtering algorithm can, for example, reduce the required power consumption of the implementation. A low power consumption is important in wireless applications, particularly at the mobile terminal side, where the physical size of the mobile terminal and long battery life are crucial. In addition, a high sampling rate limits the time for the chosen digital-signal processor to execute the operations involved. Therefore, it is desirable to choose an algorithm that with a specification on the convergence speed, has a low computational complexity without compromising final misadjustment. The algorithms derived in the thesis could

serve as viable low-complexity alternatives to the RLS algorithm in applications where a simple algorithm such as the NLMS algorithm does not fulfill the requirements set on the convergence speed and misadjustment.

The linearly-constrained adaptation algorithms derived in the thesis provide a simple mechanism to trade off computational complexity and convergence speed. The constrained affine-projection (CAP) algorithm can overcome the slow convergence of the normalized constrained LMS (NCLMS) algorithm, in situations where the input signal is colored, by increasing the number of data-reuses, i.e., using more information in each update. The increased number of data-reuses of the CAP algorithm will slightly increase the final misadjustment. The set-membership constrained affine-projection (SM-CAP) algorithm reduces the effect of this tradeoff of complexity and misadjustment. Furthermore, the reduced average computational complexity of the SM-CAP algorithm can be considerable as compared to that of the CAP algorithm, especially when the number of data-reuses increases and the CAP update is computationally complex.

The Householder Transform (HT) approach to linearly constrained problems results in an efficient implementation of LCMV adaptive filters. By a HT transformation of the input signal, the dimension of the subspace in which the coefficients of the adaptive filter are adapted can be reduced. In terms of computational complexity, the proposed Householder structure compares to the most efficient implementation of the generalized sidelobe canceler (GSC), which is a structure frequently used in literature. As with the GSC structure, the HT implementation can be used with any unconstrained adaptation algorithm.

Table 8.1 summarizes the tradeoff between the complexity and convergence of the adaptive filtering algorithms for linearly-constrained problems considered in this thesis. The comparison considers the case of colored input signals, which is the case when the most notable differences in terms of convergence rate can be made. It should be noted that for the SM-CAP algorithm, the data-selectivity of the algorithm can decrease the average complexity considerably, although the computational complexity per update can be rather high.

For the proposed family of training-based set-membership normalized algorithms with

Table 8.1: Algorithm Comparisons for LCMV Adaptive Filters.

| ALG | COMPLEXITY | CONVERGENCE |
|--------------|----------------------------|-------------------------|
| CLMS | LOW | SLOW |
| NCLMS/NHCLMS | LOW | SLOW |
| HCLMS | LOW | SLOW |
| HCAP/CAP | LOW–HIGH ^{1), 2)} | SLOW–FAST ³⁾ |
| SM-CAP | LOW–HIGH ^{1), 2)} | SLOW–FAST ³⁾ |

- 1) Complexity depends on the number of data reuses.
- 2) Reduced average complexity.
- 3) Slow convergence for low complexity and fast convergence for high complexity.

data-reuse, the computational complexity per update is easily varied. The results indicate that the algorithms can provide favorable results in terms of convergence speed and steady-state MSE, unlike the most widely used algorithms such as the LMS and NLMS algorithms. In addition, the data-selective updating of the proposed algorithms can substantially reduce the overall computational complexity as compared to the application of their conventional counterparts. The set-membership binormalized data-reusing (SM-BNDRLMS) algorithms, having two data-reuses, are viable low-complexity alternatives to the set-membership NLMS (SM-NLMS) algorithm when the input signal is colored. If the SM-BNDRLMS algorithms do not present sufficient convergence speed, an arbitrary number of data-reuses can be utilized in the coefficient update by applying the set-membership affine-projection (SM-AP) algorithm.

Adaptive set-membership filters with data-reuse update more frequently during the transient. As the solution approaches the steady-state, updating becomes less frequent. If there is a tight constraint on processing power such that a high order of the adaptive filter is needed, even the application of the SM-NLMS algorithm can turn out impractical. The new set-membership partial-update NLMS (SM-PU-NLMS) algorithm shows a new perspective for adaptive filtering when a very large number of coefficients needs to be used, and high performance needs to be maintained. The SM-PU-NLMS algorithm can be properly designed to control the computational complexity during the transient phase by

Table 8.2: Algorithm Comparisons for Training-Based Adaptive Filters.

| ALG | COMPLEXITY | CONVERGENCE |
|------------|----------------------------|-------------------------|
| PU-NLMS | LOW | SLOW |
| SM-PU-NLMS | LOW ²⁾ | SLOW |
| SM-NLMS | LOW ²⁾ | SLOW |
| SM-BNDRLMS | LOW ²⁾ | MEDIUM |
| SM-AP | LOW–HIGH ^{1), 2)} | SLOW–FAST ³⁾ |

¹⁾ Complexity depends on the number of data reuses.

²⁾ Reduced average complexity.

³⁾ Slow convergence for low complexity and fast convergence for high complexity.

only updating a fraction of all the filter coefficients. When approaching the steady-state, the number of coefficients used in the update can be further reduced. The SM-PU-NLMS offers the advantage of sparse update in time as compared to the conventional partial-update NLMS (PU-NLMS) algorithm. Furthermore, the data-dependent step size offers fast convergence and low misadjustment.

Finally, Table 8.2 summarizes the tradeoff between the complexity and convergence of the training-based adaptive filtering algorithms considered in this thesis.

As was the case for the comparison in Table 8.1, the input signals are assumed to be correlated. We note that for the case of PU algorithms, the convergence rate can in general be rather slow if only a small fraction of the filter coefficients are used for the update. However, one application where this type of algorithm can provide favorable results is network echo cancellation, where only a small fraction of the echo path to be identified is nonzero. In such cases the convergence speed may be sufficient even when a small number of coefficients are used in the update. Furthermore, the data-selective versions can provide a low overall complexity, although, the complexity per update is rather high.

8.2 Future Work

For the linearly-constrained adaptive filters considered in the thesis, the set of equations specifying the constraints was assumed to be fixed and perfectly known throughout the

adaptation. In some applications this may not hold true for all time instants. For example, we may have a mismatch in the vectors building the constraint matrix as compared to the true ones, or the constraint values and the constraint matrix could be time-varying. The case of constraint mismatch can cause signal cancellation of the signal of interest. Algorithms more robust to mismatch in the constraint matrix could be obtained if quadratic constraints are incorporated into the solution. This should be investigated for the proposed linearly-constrained adaptive filtering algorithms. If constraint values are time-varying, convergence problems may occur because the optimal solution will change accordingly. An approach to handle situations of time-varying constraints was proposed for the GSC structure in [92], where a transformation was applied to the input signal such that the adaptation was not affected by these changes. It would be of interest to investigate how this solution would combine with the Householder structure considered in this thesis.

The set-membership adaptive filtering algorithms considered herein made use of an objective function, which in some sense minimizes the Euclidean distance to the set of feasible solutions. The form of the objective function constrained the solutions to be bounded by a spheroid as was also the case for the SM-NLMS algorithm. An interesting extension of our work is to associate bounding ellipsoids to the normalized data-reusing algorithms. This may result in a family of data-reusing algorithms which can be related to the optimal bounding ellipsoid (OBE) algorithms proposed within the SMF framework [70].

The convergence analyses of the PU-NLMS and the SM-PU-NLMS algorithms were performed for white input signals. This should be extended to the case of colored input signals. Furthermore, an extension to incorporate data-reuse into the partial-update solution could improve the convergence speed for colored inputs. The expected increase in computational complexity per update as a result of reusing past data motivates an SMF approach to reduce the overall complexity.

Bibliography

- [1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, Kluwer Academic Publishers, Boston, 1997.
- [2] G.-O. Glentis, K. Berberidis, and S. Theodoridis, “Efficient least squares adaptive algorithms for FIR transversal filtering,” *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 13–41, July 1999.
- [3] S. U. Qureshi, “Adaptive equalization,” *Proceedings of the IEEE*, vol. 73, no. 9, pp. 1349–1387, September 1985.
- [4] U. Madhow and M. L. Honig, “MMSE interference suppression for direct-sequence spread-spectrum CDMA,” *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3178–3188, December 1994.
- [5] G. B. Giannakis and C. Tepedelenlioğlu, “Basis expansion models and diversity techniques for blind identification and equalization of time-varying channels,” *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1969–1986, October 1998.
- [6] L. C. Godara, “Applications of antenna arrays to mobile communications, part I: performance improvement, feasibility, and system considerations,” *Proceedings of the IEEE*, vol. 60, no. 7, pp. 1031–1060, July 1997.
- [7] G. Woodward and B. S. Vucetic, “Adaptive detection for DS-CDMA,” *Proceedings of the IEEE*, vol. 46, no. 7, pp. 1413–1434, July 1998.
- [8] D. Gesbert and P. Duhamel, “Unbiased blind adaptive channel identification,” *IEEE Transactions on Signal Processing*, vol. 48, no. 1, pp. 148–158, January 2000.

- [9] C. Breining *et. al.*, “Acoustic echo control: An application of very-high-order adaptive filters,” *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 42–69, July 1999.
- [10] S. L. Gay and J. Benesty, Eds., *Acoustic Signal Processing for Telecommunication*, Kluwer Academic Publishers, Boston, 2000.
- [11] P. Regalia, *Adaptive IIR filtering in signal processing and control*, Marcel Dekker, Inc., New York, 1995.
- [12] S. Haykin, *Adaptive Filter Theory*, New Jersey: Prentice-Hall, Englewood-Cliffs, 1996.
- [13] S L. Gay and S. Tavathia, “The fast affine projection algorithm,” *International Conference of Acoustics, Speech, and Signal Processing*, pp. 3023–3026, Detroit, USA, May 1995.
- [14] K. Doğançay and O. Tanrikulu, “Adaptive filtering algorithms with selective partial update,” *IEEE Transaction Circuits and Systems-II*, vol. 48, no. 8, pp. 762–769, August 2001.
- [15] S. M. Kuo and D. R. Morgan, “Active noise control: a tutorial review,” *Proceedings of the IEEE*, vol. 87, no. 6, pp. 943–975, June 1999.
- [16] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.
- [17] D. P. Taylor, G. M. Vitetta, B. D. Hart, and A. Mämmelä, “Wireless channel equalization,” *European Transactions on Telecommunications*, vol. 9, no. 2, pp. 117–143, Mars/April 1998.
- [18] H. Holma and A. Toskala, *Wideband CDMA for UMTS*, John Wiley and Sons, New York, 2000.
- [19] M. Latva-aho, *Advanced receivers for wideband CDMA system*, Doctoral thesis, Department of Electrical Engineering, University of Oulu, Finland, 1998.

- [20] S. Werner and J. Lilleberg, “Downlink channel decorrelation in CDMA systems with long codes,” *IEEE International Vehicular Technology Conference, VTC’99*, pp. 1614–1617, Houston, USA, May 1999.
- [21] P. Komulainen and M. Heikkilä, “Adaptive channel equalization based on chip separation for CDMA downlink,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Vehicular Technology Conference, PIMRC’99, Osaka, Japan*, pp. 978–982, September 1999.
- [22] K. Hooli, M. Juntti, and M. Latva-aho, “Inter-path interference suppression in WCDMA systems with low spreading factors,” *IEEE International Vehicular Technology Conference, VTC’99*, pp. 421–425, Amsterdam, Netherlands, November 1999.
- [23] P. B. Rapajic and B. S. Vucetic, “Adaptive receiver structures for asynchronous CDMA systems,” *IEEE Journal Selected Areas in Communications*, vol. 12, no. 4, pp. 685–697, May 1994.
- [24] M. Juntti, *Multiuser Demodulation for DS-CDMA Systems in Fading Channels*, Doctoral thesis, Department of Electrical Engineering, University of Oulu, Finland, 1997.
- [25] M. Honig, U. Madhow, and S. Verdú, “Blind multiuser detection,” *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 944–960, July 1995.
- [26] S. L. Miller, “An adaptive direct-sequence code-division multiple access receiver for multiuser interference rejection,” *IEEE Transactions on Communications*, vol. 43, no. 2, pp. 1745–1755, February–April 1995.
- [27] O. L. Frost III, “An algorithm for linearly constrained adaptive array processing,” *Proceedings of IEEE*, vol. 60, no. 8, pp. 926–935, August 1972.
- [28] L. J. Griffiths and C. W. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Transactions on Antennas and Propagation*, vol. AP-30, no. 1, pp. 27–34, January 1982.

- [29] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 5, no. 2, pp. 4–24, April 1988.
- [30] J. B. Schodorf and D. B. Williams, "Array processing techniques for multiuser detection," *IEEE Transactions on Communications*, vol. 45, no. 11, pp. 1375–1378, November 1997.
- [31] M. K. Tsatsanis and Z. Xu, "Performance analysis of minimum variance CDMA receivers," *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 3014–3022, November 1998.
- [32] U. Madhow, "Blind adaptive interference suppression for direct-sequence CDMA," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 2049–2069, October 1998.
- [33] Michael Honig and Michail K. Tsatsanis, "Adaptive techniques for multiuser CDMA receivers," *IEEE Signal Processing Magazine*, vol. 17, no. 3, pp. 49–61, May 2000.
- [34] Z. Xu and M. K. Tsatsanis, "Blind adaptive algorithms for minimum variance CDMA receivers," *IEEE Transactions on Communications*, vol. 49, no. 1, pp. 180–194, January 2001.
- [35] Z. Tian, K. L. Bell, and H. L. Van Trees, "Robust constrained linear receivers for CDMA wireless systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 7, pp. 1510–1522, July 2001.
- [36] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "Simplified FLS algorithm for linear phase adaptive filtering," *European Signal Processing Conference*, vol. 3, pp. 1237–1240, Rhodes, Greece, 1998.
- [37] Y. Chu and W.-H. Fang, "A novel Wavevelet-based generalized sidelobe canceller," *IEEE Transactions on Antennas and Propagation*, vol. 47, no. +, pp. 1485–1494, September 1999.

- [38] B. Widrow and M. E. Hoff, “Adaptive switching circuits,” *IRE Western Electric Show and Convention Record*, pp. 96–104, August 1960.
- [39] J. I. Nagumo and A. Noda, “A learning method for system identification,” *IEEE Transactions on Automatic Control*, vol. AC-12, pp. 282–287, June 1967.
- [40] R. L. Plackett, “Some theorems in least-squares,” *Biometrika*, vol. 37, pp. 149–157, 1950.
- [41] A. H. Sayed and T. Kailath, “A state-space approach to adaptive RLS filtering,” *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–16, July 1994.
- [42] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electronics and Communications in Japan*, vol. 67-A, pp. 126–132, 1984.
- [43] M. L. R. de Campos and A. Antoniou, “A new quasi-Newton adaptive filtering algorithm,” *IEEE Transactions on Circuits and Systems — Part II*, vol. 44, no. 11, pp. 924–934, November 1997.
- [44] D. T. M. Slock, “On the convergence behavior of the LMS and the normalized LMS algorithms,” *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2811–2825, September 1993.
- [45] G. C. Goodwin and S. K. Sin, *Adaptive Filtering Prediction and Control*, New Jersey: Prentice-Hall, Englewood-Cliffs, 1984.
- [46] J. A. Apolinário Jr., *New algorithms of adaptive filtering: LMS with data-reusing and fast RLS based on QR decomposition*, Dr. Sc. Thesis, COPPE/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 1998.
- [47] D. D. Falconer L. Ljung, “Application of fast Kalman estimation to adaptive equalization,” *IEEE Transactions on Communications*, vol. 26, pp. 1439–1446, October 1978.

- [48] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, “A fast sequential algorithm for least-squares filtering and prediction,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-31, pp. 1394–1402, December 1983.
- [49] J. M. Cioffi and T. Kailath, “Fast recursive-least-squares transversal filters for adaptive filters,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-32, pp. 304–337, April 1984.
- [50] R. Merched and A. H. Sayed, “Extended fast fixed-order RLS adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3015–3031, December 2001.
- [51] J. A. Apolinário, M. L. R. de Campos, and P. S. R. Diniz, “Convergence analysis of the binormalized data-reusing LMS algorithm,” *IEEE Transaction on Signal Processing*, vol. 48, no. 11, pp. 3235–3242, November 2000.
- [52] J. A. Apolinário, M. L. R. de Campos, P. S. R. Diniz, and T. I. Laakso, “Binormalized data-reusing LMS algorithm with optimized step-size sequence,” *Brazilian Telecommunications Journal - SBT*, vol. 13, pp. 49–55, 1998.
- [53] B. A. Schnaufer, *Practical Techniques for rapid and reliable real-time adaptive filtering*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA, 1995.
- [54] S. G. Sankaran and A. A. Beex, “Convergence behavior of affine projection algorithms,” *IEEE Transactions on Signal Processing*, vol. 48, no. 4, pp. 1086–1096, April 2000.
- [55] D. T. M. Slock, “The block underdetermined covariance BUC fast transversal filter (FTF) algorithm for adaptive filtering,” *Twenty-Second Asilomar Conference on Signals, System and Computers*, pp. 550–554, September Pacific Grove, CA, USA, October 1992.

- [56] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 444–453, February 1995.
- [57] M. Rupp, "A family of adaptive filtering algorithms with decorrelating properties," *IEEE Transactions on Signal Processing*, vol. 46, no. 3, pp. 771–775, March 1998.
- [58] M. L. R. de Campos, J. A. Apolinário, and P. S. R. Diniz, "On normalized data-reusing and affine projection algorithms," *IEEE International Conference on Electronics, Circuits and Systems, ICECS'99*, pp. 843–846, Pafos, Cyprus, September 1999.
- [59] M. L. R. de Campos, *Development and analysis of fast and robust Newton-type adaptation algorithms*, Ph.D. Thesis, University of Victoria, Canada, 1995.
- [60] M. L. R. de Campos and A. Antoniou, "Analysis of a new quasi-Newton adaptive filtering algorithm," *IEEE International Conference on Electronics, Circuits, Systems*, vol. 44, pp. 924–934, Rhodes Greece, 1996.
- [61] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing*, PTR: Prentice Hall, Englewood Cliffs, NJ, 1993.
- [62] M. H. Hayes, *Statistical Digital Signal Processing and Modelling*, John Wiley and Sons, New York, 1996.
- [63] T. C. Liu and B. D. Van Veen, "Modular structure for implementation of linearly constrained variance beamformers," *IEEE Transactions on Signal Processing*, vol. 39, no. 10, pp. 2343–2346, October 1991.
- [64] R. Sharma and B. D. Van Veen, "Large modular structures for adaptive beamforming and the Gram-Schmidt preprocessor," *IEEE Transactions on Signal Processing*, vol. 42, no. 2, pp. 448–451, February 1994.

- [65] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A multistage representation of the Wiener filter based on orthogonal projections," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2943–2959, November 1998.
- [66] M. L. R. de Campos, S. Werner, and J. A. Apolinário, "Householder-transform constrained LMS algorithms with reduced-rank updating," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. IV, pp. 1857–1860, Phoenix USA, March 1999.
- [67] M. L. R. de Campos, S. Werner, and J. A. Apolinário, "Constrained adaptation algorithms employing Householder transformation," *IEEE Transaction on Signal Processing*, vol. 50, pp. 2187–2195, September 2002.
- [68] S. Gollamudi, S. Nagaraj, and Y. F. Huang, "SMART: A toolbox for set-membership filtering," *European Conference on Circuit Theory and Design*, pp. 879–884, August 1997.
- [69] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, no. 5, pp. 111–114, May 1998.
- [70] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. F. Huang, "Set-membership adaptive equalization and updatator-shared implementation for multiple channel communications systems," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2372–2384, September 1998.
- [71] S. Nagaraj, S. Gollamudi, S. Kapoor, Y. F. Huang, and J. R. Deller, "Adaptive interference suppression for CDMA systems with a worst-case error criterion," *IEEE Transactions on Signal Processing*, vol. 48, no. 1, pp. 284–289, January 2000.
- [72] J. R. Deller, S. Gollamudi, S. Nagaraj, and Y. F. Huang, "Convergence analysis of the QUASI-OBE algorithm and the performance implications," *IFAC International Symposium on System Identification, SYSID'2000*, vol. 3, pp. 875–880, Santa Barbara, USA, June 2000.

- [73] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Transaction Circuits and Systems-II*, vol. 44, no. 3, pp. 209–216, March 1997.
- [74] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Transaction on Signal Processing*, vol. 47, no. 5, pp. 1421–1427, May 1999.
- [75] K. Doğançay and O. Tanrikulu, "Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'2000*, vol. 1, pp. 448–451, Istanbul, Turkey, June 2000.
- [76] S. Werner, M. L. R. de Campos, and J. A. Apolinário, "On the equivalence of the constrained rls and the gsc-rls beamformers," *IEEE International Telecommunications Symposium, ITS2002*, Natal, Brazil, September 2002.
- [77] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing algorithms," *IFAC Symposium on System Identification, SYSID'2000*, vol. 3, pp. 869–874, Santa Barbara, California, USA, June 2000.
- [78] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing algorithms," *IEEE Transactions on Signal Processing*, 2002 (to appear).
- [79] S. Werner, M. L. R. de Campos, and J. A. Apolinário, "Data-selective constrained affine projection algorithm," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'01*, vol. VI, pp. 3745–3748, Salt Lake City, Utah, USA, May 2001.
- [80] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update nlms algorithms with data-selective updating," *submitted to IEEE Transaction on Signal Processing*, 2002.

- [81] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, “Partial-update nlms algorithms with data-selective updating,” *IEEE International Conference of Acoustics, Speech, and Signal Processing*, Orlando, Florida, USA, May 2002.
- [82] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, “Mean-squared analysis of the partial-update nlms algorithm,” *IEEE International Telecommunications Symposium, ITS2002*, Natal, Brazil, September 2002.
- [83] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, “Adaptive antenna systems,” *Proceedings of the IEEE*, vol. 55, pp. 2143–2159, December 1967.
- [84] L. J. Griffiths, “A simple adaptive algorithm for real-time processing in antenna arrays,” *Proceedings of the IEEE*, vol. 57, pp. 1696–1704, October 1969.
- [85] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, “A fast least-squares algorithm for linearly constrained adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1168–1174, May 1996.
- [86] J. A. Apolinário, S. Werner, and P. S. R. Diniz, “Constrained normalized adaptive filters for CDMA mobile communications,” *European Signal Processing Conference*, vol. 4, pp. 2053–2056, Rhodes, Greece, 1998.
- [87] M. L. R. de Campos, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, “Constrained quasi-Newton algorithm for CDMA mobile communications,” *International Telecommunications Symposium*, pp. 371–376, São Paulo, Brazil, August 1998.
- [88] M. L. R. de Campos J. A. Apolinário and P. S. R. Diniz, “The constrained affine projection algorithm – development and convergence issues,” *First Balkan Conference on Signal Processing, Communications, Circuits and Systems*, Istanbul, Turkey, May 2000.
- [89] L. S. Resende, C. A. F. Rocha, and M. G. Bellanger, “A linearly-constrained approach to the interpolated FIR filtering problem,” *Proc. IEEE International Conference on*

- Acoustics, Speech, and Signal Processing, ICASSP'00*, vol. 1, pp. 392–395, Istanbul, Turkey, June 2000.
- [90] S. C. Park and J. F. Doherty, “Generalized projection algorithm for blind interference suppression in DS/CDMA communications,” *IEEE Transactions Circuits and Systems — Part II*, vol. 44, no. 5, pp. 453–460, June 1997.
- [91] M. T. Schiavoni and M. G. Amin, “A linearly constrained minimization approach to adaptive linear phase and notch filters,” *Proceedings of the Twentieth Southeastern Symposium on System Theory*, pp. 682–685, Philadelphia, PA, USA, March 1988.
- [92] C.-Y. Tseng and L. J. Griffiths, “A systematic procedure for implementing the blocking matrix in decomposed form,” *Twenty-Second Asilomar Conference on Signals, System and Computers*, vol. 2, pp. 808–812, California, USA, October 1988.
- [93] S.-F. Hsiao and J.-M. Delosme, “Householder CORDIC algorithms,” *IEEE Transactions on Computers*, vol. 44, no. 8, pp. 990–1001, August 1995.
- [94] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Baltimore: The Johns Hopkins University Press, 1983.
- [95] A. O. Steinhardt, “Householder transforms in signal processing,” *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 5, no. 3, pp. 4–12, July 1988.
- [96] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, pub-OUP, 1995.
- [97] C. A. Medina, C. V. Rodríguez, and J. A. Apolinário, “On the use of Householder transformation in adaptive microphone array,” *Proc. 4th World Multiconference on Systems, Cybernetics, and Informatics*, Orlando, USA, July 2000.
- [98] S. Gollamudi, *Nonlinear mean-square multiuser detection and set-theoretic estimation for digital communications*, Ph.D. Thesis, University of Notre Dame, USA, 2000.
- [99] S. Nagaraj, *Adaptive multiuser signal estimation and downlink beamforming for wireless communications*, Ph.D. Thesis, University of Notre Dame, USA, 2000.

- [100] E. Fogel and Y. F. Huang, "On the value of information in system identification - bounded noise case," *Automatica*, vol. 18, no. 2, pp. 229–238, 1982.
- [101] J. R. Deller, "Set-membership identification in digital signal processing," *IEEE ASSSP Magazine*, vol. 6, pp. 4–22, October 1989.
- [102] J. R. Deller, M. Nayeri, and S. F. Odeh, "Least-squares identification with error bounds for real-time signal processing and control," *Proceedings of IEEE*, vol. 81, no. 6, pp. 813–849, June 1993.
- [103] E. Walter and H. Piet-Lahanier, "Estimation of parameter bounds from bounded-error: A survey," *Mathematics and Computers in Simulation*, vol. 32, no. 5–6, pp. 449–468, December 1990.
- [104] S. Dasgupta and Y. F. Huang, "Asymptotically convergent modified recursive least-squares with data dependent updating and forgetting factor for systems with bounded noise," *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 383–392, May 1987.
- [105] J. R. Deller, M. Nayeri, and M. Liu, "Unifying the landmark developments in optimal bounding ellipsoid identification," *International Journal on Adaptive Control and Signal Processing*, vol. 8, pp. 43–60, January–February 1994.
- [106] D. Joachim, J. R. Deller, and M. Nayeri, "Multiweight optimization in OBE algorithms for improved tracking and adaptive identification," *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98*, vol. 4, pp. 2201–2204, Seattle, USA, May 1998.
- [107] S. Kapoor, S. Gollamudi, S. Nagaraj, and Y. F. Huang, "Adaptive multiuser detection and beamforming for interference suppression in CDMA mobile radio systems," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1341–1355, September 1999.

- [108] S. Nagaraj, S. Gollamudi, S. Kapoor, and Y. F. Huang, “BEACON: An adaptive set-membership filtering technique with sparse updates,” *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 2928–2941, November 1999.
- [109] S. Gollamudi, S. Nagaraj, and Y. F. Huang, “Blind equalization with a deterministic constant modulus cost—a set-membership filtering approach,” *International Conference of Acoustics, Speech, and Signal Processing, ICASSP’00*, vol. 5, pp. 2765–2768, Istanbul, Turkey, June 2000.
- [110] J. Panoff, S. Nagaraj, S. Gollamudi, Y.-F. Huang, and J. A. Nossek, “A minimax approach to open-loop downlink beamforming,” *Proc. IEEE International Symposium on Circuits and Systems, ISCAS’01*, vol. 2, pp. 73–76, Sydney, Australia, May 2001.
- [111] S. L. Gay, *Fast Projection Algorithms with Application to Voice Echo Cancellation*, Ph.D. Thesis, Rutgers The State University of New Jersey, New Brunswick, New Jersey, 1994.
- [112] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, “Fast projection algorithm and its step size control,” *Proceedings International Conference of Acoustics, Speech, and Signal Processing, ICASSP’95*, pp. 945–948, Detroit, MI, USA, May 1995.
- [113] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, 2001.
- [114] S. C. Douglas, “A family of normalized LMS algorithms,” *IEEE Signal Processing Letters*, vol. 1, no. 3, pp. 49–51, March 1994.
- [115] S. C. Douglas, “Analysis and implementation of the max-NLMS adaptive filter,” *Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 659–663, Pacific Grove, CA, USA, October, 1995.
- [116] M. I. Haddad, K. Mayyas, and M. A. Khasawneh, “Analytical development of the MMAXNLMS algorithm,” *Proc. IEEE International Conference on Acoustics,*

- Speech, and Signal Processing, ICASSP'99*, vol. 4, pp. 1853–1856, Phoenix, USA, March 1999.
- [117] M. Godavarti and A. O. Hero III, “Stochastic partial update LMS algorithm for adaptive arrays,” *IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 322–326, Ann Arbor, MI, USA, March 2000.
- [118] T. Schertler, “Selective block update of NLMS type algorithms,” *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, vol. 3, pp. 1717–1720, Seattle, USA, May 1998.
- [119] S. Attallah and S. W. Liaw, “DCTLMS algorithm employing partial coefficient updates,” *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 218–223, Lake Louise, Alta., Canada, October 2000.
- [120] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*, McGraw-Hill, New York, 1989.
- [121] J. Pitas, “Fast algorithms for running ordering and max/min calculation,” *IEEE Transaction Circuits and Systems-II*, vol. 36, no. 6, pp. 795–804, June 1989.
- [122] H. J. Godwin, “Some low moments of order statistics,” *Annals of Mathematical Statistics*, vol. 20, no. 2, pp. 279–285, June 1949.
- [123] H. L. Harter, “Expected values of normal order-statistics,” *Biometrika*, vol. 48, no. 1–2, pp. 151–165, June 1961.
- [124] Z. Govindarajulu, “Exact moments of order statistics in samples from the chi-distribution (1 d.f.),” *Annals of Mathematical Statistics*, vol. 33, no. 4, pp. 1292–1305, December 1963.
- [125] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1991.

VITA

Surname: Werner

Given Names: Stefan Anders

Place of Birth: Köping, Sweden

Degrees Awarded:

| | | |
|------------------------------------|-------------------------------|------|
| M.Sc. | Royal Institute of Technology | 1998 |
| Univ.Cert. Electr. Comput. Science | Royal Institute of Technology | 1992 |

Publications:

1. P. S. R. Diniz, S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Transactions on Signal Processing*, 2002 (accepted).
2. M. L. R. de Campos, S. Werner, J. A. Apolinário Jr., "Constrained adaptation algorithms employing Householder transformation," *IEEE Transaction on Signal Processing*, vol. 50, pp. 2187–2195, September 2002.
3. S. Werner, M. L. R. de Campos, P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," submitted to *IEEE Transaction on Signal Processing*, March 2002.
4. S. Werner, J. A. Apolinário Jr., M. L. R. de Campos, "On the equivalence of the constrained RLS and the GSC-RLS beamformers," *IEEE International Telecommunications Symposium, ITS'2002*, Natal, Brazil, September 2002.
5. S. Werner, M. L. R. de Campos, P. S. R. Diniz, "Mean-squared analysis of the partial-update NLMS algorithm," *IEEE International Telecommunications Symposium, ITS'2002*, Natal, Brazil, September 2002.
6. S. Werner, M. L. R. de Campos, P. S. R. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'02*, pp. 1413–1416, Orlando, Florida, USA, May 2002.
7. S. Werner, P. S. R. Diniz, "Set-membership affine projection algorithm," *Signal Processing Letters*, vol. 8, no. 8, pp. 231–235, 2001.
8. S. Werner, M. L. R. de Campos, J. A. Apolinário Jr., "Data-selective constrained affine projection algorithm," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'01*, vol. VI, pp. 3745–3748, Salt Lake City, Utah, USA, May 2001.
9. J. A. Apolinário Jr., S. Werner, P. S. R. Diniz, "Conjugate gradient algorithm with data selective updating," *SBT/Brazilian Telecommunications Symposium*, Fortaleza, Brazil, September 2001.

10. P. S. R. Diniz, S. Werner, "Set-membership binormalized data-reusing algorithms," *IFAC Symposium on System Identification, SYSID'2000*, vol. 3, pp. 869–874, Santa Barbara, California, USA, June 2000.
11. P. S. R. Diniz, S. Werner, "Adaptive Systems in Wireless Communications," *First IEEE Balkan Conference on Signal Processing, Communications, Circuits, and Systems*, Istanbul, Turkey, pp. 1-4, CDROM, June 2000.
12. S. Werner, M. L. R. de Campos, J. A. Apolinário Jr., "Kalman-Filter Based Chip Estimator for WCDMA Downlink Detector," *European Signal Processing Conference, EUSIPCO'00*, Tampere, Finland, Sept. 2000.
13. S. Werner, J. Lilleberg, "Downlink channel decorrelation in CDMA systems with long codes," *IEEE International Vehicular Technology Conference, VTC'99*, pp. 1614–1617, Houston, USA, May 1999.
14. M. L. R. de Campos, S. Werner, J. A. Apolinário Jr., "Householder-transform constrained LMS algorithms with reduced-rank updating," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'99*, vol. IV, pp. 1857–1860, Phoenix USA, March 1999.
15. J. A. Apolinário Jr., S. Werner, P. S. R. Diniz, T. I. Laakso, "Constrained normalized adaptive filters for CDMA mobile communications," *European Signal Processing Conference, EUSIPCO'98*, vol. 4, pp. 2053–2056, Rhodes, Greece, 1998.
16. R. Baghaie, S. Werner, T. Laakso, "Relaxed look-ahead techniques for pipelined implementation of adaptive multiple-antenna CDMA mobile receivers," *European Signal Processing Conference, EUSIPCO'98*, Rhodes, Greece, 1998.
17. M. L. R. de Campos, S. Werner, J. A. Apolinário Jr., T. I. Laakso, "Constrained quasi-Newton algorithm for CDMA mobile communications," *IEEE International Telecommunications Symposium, ITS'98*, pp. 371–376, São Paulo, Brazil, August 1998.
18. S. Werner, J. A. Apolinário Jr., "Multiple-antenna CDMA mobile reception using constrained normalized adaptive algorithms," *IEEE International Telecommunications Symposium, ITS'98*, pp. 354–358, São Paulo, Brazil, August 1998.
19. S. Werner, T. Laakso, J. Lilleberg, "Multiple-antenna receiver for CDMA mobile reception," *IEEE International Conference on Communications, ICC'98*, pp. 1053–1057, Atlanta, GA, USA, June 1998.
20. S. Werner, J. A. Apolinário Jr., "Application of constrained normalized algorithms for a multiple-antenna CDMA mobile receiver," *IEEE Nordic Signal Processing Symposium, NORSIG'98*, Vigsø, Denmark, June 1998.
21. R. Baghaie, S. Werner, "Pipelined adaptive CDMA mobile receivers," *IEEE Nordic Signal Processing Symposium, NORSIG'98*, Vigsø, Denmark, June 1998.

22. R. Baghaie, S. Werner, T. Laakso, "Pipelined implementation of adaptive multiple-antenna CDMA receivers," *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'98*, pp. 3229–3232, Seattle, WA, USA, May 1998.