

ADAPTIVE ALGORITHMS
EMPLOYING TAP SELECTION FOR
SINGLE CHANNEL AND STEREOPHONIC
ACOUSTIC ECHO CANCELLATION

ANDY W.H. KHONG

A Thesis submitted in fulfilment of requirements for the degree of
Doctor of Philosophy of University of London and
Diploma of Imperial College

Communications and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London
University of London
2006

*To my wife Daphne, my parents,
Mabel, Andrew, Natalie and*

To Patrick

Abstract

A class of adaptive algorithms employing tap selection for acoustic echo cancellation (AEC) is developed and analyzed in this thesis. The starting point of this work is the MMax normalized least-mean-square (MMax-NLMS) algorithm where only a subset of taps are selected for adaptation. The MMax tap selection is extended to the affine projection (AP) and recursive least squares (RLS) algorithms. The performances of these algorithms are studied in the context of single channel AEC by developing a generalized analysis framework for a wide range of algorithms including NLMS, AP, RLS and, in particular the MMax selective-tap algorithms. This analysis presents new insights into their tracking performances under both time-invariant and time-varying system conditions.

A novel approach to reduce interchannel coherence based on tap selection for stereophonic acoustic echo cancellation (SAEC) is introduced. This tap selection technique optimizes jointly the reduction in interchannel coherence and maximizing the “MMax-ness” of both channels. The reduction in interchannel coherence is achieved by an exclusive tap selection such that the same tap-indices may not be selected in both channels. The resultant exclusive-maximum (XM) tap selection is then applied to the NLMS, AP and RLS algorithms.

New insights into the SAEC problem are presented by deriving the relationship between interchannel coherence and conditioning of the two-channel input autocorrelation matrix. Employing this relationship, this work examines how the XM tap selection reduces the interchannel coherence and improves the conditioning of the input autocorrelation matrix to achieve a fast convergence. The XM tap selection is extended to the frequency-domain adaptive algorithms, employing both the 50% and an arbitrary overlapping factor between successive tap-input vectors. Simulation results verifying analysis and comparative results of the proposed algorithms will be provided in the context of single channel and stereophonic AEC.

Acknowledgment

This thesis is the result of three years of research and I would like to take this opportunity to thank many people who have contributed towards this work in one way or another.

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Patrick Naylor who has given me valuable guidance and sound advice throughout my stay at Imperial College London. It is through him that I was introduced to the field of adaptive algorithms and digital signal processing and throughout our meetings, he never fail to inspire me as a researcher and an academic. I am very grateful for his constant patience, enthusiasm and interest in my overall well-being throughout these years. I am thankful for the many wonderful opportunities given to me and the valuable academic experiences which I could not otherwise have gained.

I am also very grateful for Prof. Jacob Benesty of Université du Québec, INRS-EMT, who gave me the valuable opportunity to collaborate with him and hosted my three months stay in Montréal. I thank him for generously sharing his knowledge on the SAEC problem hence giving me further insights into the problem. Being concerned about simulations and mathematical derivations, he was ready for any discussions and queries that I had. I sincerely appreciate his valuable comments and advice rendered to me regarding both research and non-research related issues.

I thank the Engineering and Physical Sciences Research Council (EPSRC), UK for funding my research work. This work has also received many valuable comments and suggestions from the IEEE research community and in particular, I would like to thank Dr. Yiteng (Arden) Huang of Bell Laboratories and Dr. Futoshi Asano of National Institute of Advanced Industrial Science and Technology who organized the review process and the anonymous reviewers who have given many valuable suggestions towards this research. I

also appreciate Dennis Morgan of Bell Laboratories for pointing out the significance of the flat plate wind screen in front of the Western Electric “8-ball” microphone. I would like to thank Mr. Mike Brookes of Imperial College London and Prof. Yves Greiner of Telecom Paris, ENST, who examined and reviewed this thesis including Dr. Darren Ward who examined this work during its early stages.

I am indebted to my parents, my sister and brother for their constant support and their concern of my well-being over the years. Last but not least, I would like to express my greatest gratitude to my wife, Daphne, who has been with me through every stage of this research and it is through her continual love, support and confidence in me that I am able to complete this work.

Andy W. H. Khong

Contents

Abstract	3
Acknowledgment	4
Contents	6
List of Figures	11
List of Tables	15
Abbreviations	16
Notation	17
Chapter 1. General Introduction	21
1.1 Developments in echo cancellation	21
1.2 Research aim and thesis structure	24
1.3 Statement of originality, contributions and related publications	26
Chapter 2. Algorithms Employing Tap Selection in Single Channel Acous- tic Echo Cancellation	29
2.1 Introduction	29
2.2 The single channel acoustic echo cancellation problem	32
2.2.1 Problem definition	32
2.2.2 Assumptions	34
2.3 Performance measure	36
2.3.1 Echo return loss enhancement	36
2.3.2 Normalized misalignment	37
2.3.3 Relationship between normalized misalignment and error in single channel AEC	37
2.4 The LMS and NLMS algorithms	39
2.5 Partial update adaptive algorithms	43

2.5.1	The Periodic-LMS and Sequential-LMS algorithms	43
2.5.2	The Selective-partial-update NLMS algorithm	45
2.5.3	The Max-NLMS and MMax-NLMS algorithms	47
2.6	Adaptive algorithms employing partial updates	49
2.6.1	The MMax affine projection algorithm	49
2.6.2	The MMax recursive least squares algorithm	50
2.7	Computational complexity	53
2.8	Simulation results	55
2.8.1	Experimental setup	55
2.8.2	NLMS-based simulations	57
2.8.3	AP-based simulations	59
2.8.4	RLS-based simulations	60
2.9	Conclusions	62
2.10	Appendix	63
2.10.1	Partial updating and selective-tap algorithms	63

Chapter 3. Tracking Performance of MMax Algorithms Under Time-varying Unknown System Conditions 66

3.1	Introduction	66
3.2	Non-stationary system model	68
3.3	General misalignment analysis for time-varying systems	69
3.3.1	Mean square misalignment with $K = 1$ and $M = L$ for NLMS and RLS	74
3.3.2	Mean square misalignment with $K \neq 1$ and $M = L$ for AP	79
3.4	Misalignment analysis of algorithms employing MMax tap selection	80
3.4.1	Misalignment analysis of MMax-NLMS	80
3.4.2	Misalignment analysis of MMax-RLS	86
3.4.3	Misalignment analysis for MMax-AP	88
3.5	Simulations and results	91
3.5.1	Comparison between tracking a modified Markov model and the image model	91
3.5.2	Effect of non-stationarity on misalignment for NLMS and MMax-NLMS	92
3.5.3	Effect of non-stationarity on misalignment for AP and MMax-AP	95
3.5.4	Effect of non-stationarity on misalignment for RLS and MMax-RLS	95
3.5.5	Effect of step-size on misalignment for NLMS and MMax-NLMS	96
3.5.6	Effect of tap selection M on normalized misalignment	97

3.5.7	Effect of SNR on normalized misalignment	99
3.6	Discussion and conclusions	101
3.7	Appendix	102
3.7.1	Fourth-order factorization for zero mean gaussian variables	102
3.7.2	Verification of step-size boundary condition for MMax-NLMS using contraction mapping	103
Chapter 4. Stereophonic Acoustic Echo Cancellation Employing Tap Se-		
lection		105
4.1	Introduction	105
4.2	Problems associated with stereophonic acoustic echo cancellation	108
4.3	Tap selection for SAEC	112
4.3.1	Dependency of convergence rate on \mathcal{M}	112
4.3.2	Interchannel decorrelation using tap selection	114
4.4	Exclusive-maximum tap selection	118
4.4.1	Formulation	118
4.4.2	Efficient realization: the exclusive-maximum tap selection	123
4.5	Exclusive-maximum adaptive filtering	126
4.5.1	The XM-NLMS algorithm	127
4.5.2	The XMNL-NLMS algorithm	127
4.5.3	The XMNL-AP algorithm	128
4.5.4	The XMNL-RLS algorithm	129
4.5.5	Computational complexity	131
4.6	Simulation results	133
4.6.1	Experimental setup	133
4.6.2	NLMS-based simulations	133
4.6.3	AP-based simulations	134
4.6.4	RLS-based simulations	136
4.7	Conclusions	137
4.8	Appendix	138
4.8.1	Proof of non-unique solutions in SAEC	138
4.8.2	Verification of exclusive tap selection set which maximizes \mathcal{M}	139
4.8.3	XMNL-based algorithms for SAEC	141
Chapter 5. Frequency-Domain Algorithms with Applications to Stereo-		
phonic Acoustic Echo Cancellation		143
5.1	Introduction	143
5.2	Definition of data-sectioning and commonly used matrices	146

5.3	Single channel frequency-domain adaptive algorithms: A review	147
5.3.1	The fast-LMS adaptive algorithm	148
5.3.2	The multi-delay filtering (MDF) structure	152
5.3.3	General derivation of frequency-domain algorithms	156
5.3.4	Steady-state misalignment	166
5.4	Effect of interchannel coherence on the conditioning of \mathbf{R}_{xx} for SAEC . . .	168
5.4.1	Two-channel autocorrelation matrix and the normal equations . . .	169
5.4.2	Autocorrelation matrix and spectral content	169
5.4.3	The E-norm condition number of autocorrelation matrix	171
5.4.4	Relationship between interchannel coherence and the conditioning of \mathbf{R}_{xx}	172
5.4.5	Application to two-channel frequency-domain adaptive algorithm . .	175
5.5	Frequency-domain adaptive filtering employing XM tap selection	176
5.5.1	Effect of XM tap selection on interchannel coherence and condition number of \mathbf{R}_{xx}	177
5.5.2	Selection in frequency-domain vs selection in time-domain	178
5.5.3	Tap selection with 50% overlapping factor	181
5.5.4	Tap selection with arbitrary overlapping factor	183
5.6	Simulation results	185
5.6.1	Verification of (5.120)	185
5.6.2	Experimental setup for FLMS and FLMS α based simulations	187
5.6.3	FLMS with 50% overlapping-factor simulations	187
5.6.4	FLMS with arbitrary overlapping-factor simulations	189
5.7	Conclusions	190
5.8	Appendix	191
5.8.1	Proof of matrix multiplication $(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01}$	191
5.8.2	Proof of equation (5.53)	192
5.8.3	Proof of equation (5.77)	192
5.8.4	Proof of equation (5.78)	192
5.8.5	Explicit link between $\chi_E[\mathbf{R}_{xx}^{1/2}]$ and $\chi_F[\mathbf{R}_{xx}^{1/2}]$ for uncorrelated $x_1(n)$ and $x_2(n)$	193
5.8.6	Additional results with interchannel coherence controlled by adding WGN to channel 2	194
5.8.7	Frequency-domain algorithms	196
Chapter 6. Discussion and Conclusions		201
6.1	Summary	201

6.2	Conclusions	203
6.3	Future work	204
6.4	List of publications arising directly from this thesis	205
6.5	Other publications	207
	Bibliography	208

List of Figures

2.1	Schematic diagram of single channel acoustic echo cancellation.	33
2.2	Receiving room impulse response $\mathbf{h}(n)$ generated using the method of images at $f_s =$ 8 kHz and $L_R = 1024$	56
2.3	Normalized misalignment comparison for single channel partial update algorithms using WGN input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mathcal{N} = 2$, $B = 16$, $\mathcal{B} = 16$, $\mu_{\text{NLMS}} = 0.7$, $\mu_{\text{Periodic}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, $\mu_{\text{SPU}} = 0.6$, $\mu_{\text{Sequential}} = 0.5$, SNR = 30 dB].	57
2.4	Normalized misalignment for single channel NLMS and MMax-NLMS using WGN in- put [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{\text{NLMS}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, SNR = 30 dB].	58
2.5	Normalized misalignment for single channel NLMS and MMax-NLMS using speech in- put [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{\text{NLMS}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, SNR = 30 dB].	59
2.6	Normalized misalignment for single channel AP and MMax-AP using WGN input [$L_T =$ $L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu = 0.7$, $K = 2$, SNR = 30 dB].	60
2.7	Normalized misalignment for single channel AP and MMax-AP using speech input [$L_T =$ $L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{\text{AP}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, SNR = 30 dB].	60
2.8	Normalized misalignment for single channel RLS and MMax-RLS [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\lambda = 0.9993$, SNR = 30 dB].	61
2.9	Normalized misalignment for single channel RLS and MMax-RLS using speech in- put [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\lambda = 0.9998$, SNR = 30 dB].	61
3.1	Variation of μ_{mis} with ξ under various SNR conditions.	77
3.2	Variation of ψ with M selected coefficients per iteration for $\xi < 1$ showing a modest increment in ψ when $0.5L \leq M < L$	84
3.3	NLMS normalized misalignment for various source velocity with impulse response \mathbf{h}_n generated using the method of images [$L_R = L = 64$, $\mu = 0.1$, SNR = 40 dB].	91
3.4	NLMS normalized misalignment for various ξ with impulse response \mathbf{h}_n generated using modified Markov model [$L_R = L = 64$, $\mu = 0.1$, $\sigma_s^2 = 1$, SNR = 40 dB].	92

3.5	NLMS normalized misalignment for various ξ [$L = 64, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$].	93
3.6	MMax-NLMS normalized misalignment for various ξ [$L = 64, M = 8, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$]. Dashed lines indicate corresponding performance for NLMS. . . .	93
3.7	AP normalized misalignment for various ξ [$L = 64, K = 3, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$].	94
3.8	MMax-AP normalized misalignment for various ξ [$L = 64, M = 8, K = 3, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$]. Dashed lines indicate corresponding performance for AP.	94
3.9	RLS normalized misalignment for various ξ [$L = 64, \lambda = 0.9948, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$].	95
3.10	MMax-RLS normalized misalignment for various ξ [$L = 64, M = 4, \lambda = 0.9948, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$]. Dashed lines indicate corresponding performance for RLS.	96
3.11	NLMS: Variation of average normalized misalignment with μ [$L = 128, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 30 \text{ dB}$].	98
3.12	MMax-NLMS: Variation of average normalized misalignment with μ [$\sigma_x^2 = \sigma_s^2 = 1, L = 128, M = 64, \text{SNR} = 30 \text{ dB}$].	98
3.13	Variation of average normalized misalignment with the number of taps selected for adaptation M [$L = 32, \mu = 0.1, K = 3, \lambda = 0.9896, \xi = 0.9999, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40 \text{ dB}$].	99
3.14	MMax-NLMS and MMax-AP: Variation of average normalized misalignment with SNR [$L = 128, M = 64, \mu = 0.1, K = 3, \xi = 0.99999, \sigma_x^2 = \sigma_s^2 = 1$].	100
3.15	MMax-RLS: Variation of average normalized misalignment with SNR [$L = 128, M = 64, \lambda = 0.9974, \xi = 0.99999, \sigma_x^2 = \sigma_s^2 = 1$].	100
4.1	Schematic diagram of stereophonic acoustic echo cancellation. Only one channel of the return path is shown for simplicity.	106
4.2	Variation of \mathcal{M} with subselection parameter M showing modest reduction of \mathcal{M} within the region $0.5L \leq M < L$ for zero mean unit variance WGN sequence with $L = 256$	114
4.3	Number of iterations for MMax-NLMS to converge to -20 dB normalized misalignment, as a function of \mathcal{M} for $L = 256$	114
4.4	Squared interchannel coherence for (a) $M = L = 512$, (b) $M = 0.5L$ with MMax tap selection and (c) $M = 0.5L$ with exclusive tap selection.	115
4.5	Effect of exclusive tap selection for various mean value of $ \gamma(f) ^2$ across frequency on mean condition number for zero mean unit variance WGN sequence (a) without tap selection, (b) with MMax tap selection and (c) with exclusive tap selection.	117
4.6	Possible combinations of tap selection for $L = 4$ and $M = 2$ in channels 1 and 2.	119
4.7	Cost matrices for \mathcal{A} and \mathcal{C}	120

4.8	Misalignment for (a) $\vartheta_m = 1$, (b) NLMS, (c) $\vartheta_m = 0.9$, (d) $\vartheta_m = 0.7$, (e) $\vartheta_m = 0.1$ [$L = 6$, $M = 3$, $\mu = 0.6$, $\epsilon = 0.9$, SNR = 40 dB].	121
4.9	Number of samples that a particular integer cost in matrices (a) \mathcal{A} and (b) \mathcal{C} associated with ζ_{\min} is selected for adaptation [$L = 6$, $M = 3$, $\vartheta_m = 0.1$, $\mu = 0.6$, $\epsilon = 0.9$, SNR = 40 dB].	122
4.10	Schematic diagram of the XMNL preprocessor in stereophonic acoustic echo canceller. Bold arrows indicate tap selection control.	127
4.11	Computational complexity of NLMS and AP-based algorithms.	132
4.12	Computational complexity of RLS-based algorithms.	132
4.13	Normalized Misalignment for WGN sequence (a) NLMS, (b) NL-NLMS (c) XM-NLMS and (d) XMNL-NLMS [$L_T = 1200$, $L_R = 256$, $L = 256$, $\mu = 0.1$, $\beta = 0.5$, SNR = 25 dB].	133
4.14	Normalized misalignment using WGN input sequence for (a) NL-AP and (b) XMNL- AP [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-AP}} = 0.15$, $\mu_{\text{XMNL-AP}} = 0.1$, $\beta = 0.5$, $K = 2$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	134
4.15	Normalized misalignment using (a) speech input sequence for (b) NL-AP and (c) XMNL- AP [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\mu_{\text{NL-AP}} = 0.15$, $\mu_{\text{XMNL-AP}} = 0.1$, $\beta = 0.5$, $K = 2$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	135
4.16	Normalized Misalignment with unit variance WGN source for (a) NL-RLS and (b) XMNL- RLS [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\lambda_{\text{XMNL-RLS}} = 0.9987$, $\lambda_{\text{NL-RLS}} = 0.9975$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	136
4.17	(a) Speech signal and normalized misalignment with unit variance WGN source for (b) NL-RLS and (b) XMNL-RLS [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\lambda_{\text{XMNL-RLS}} =$ 0.99961 , $\lambda_{\text{NL-RLS}} = 0.99957$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	137
5.1	Schematic of the FLMS algorithm (after [115]).	148
5.2	Input sequence partitioning for the FLMS algorithm with 50% overlap between successive frames.	149
5.3	Schematic of the MDF structure (after [23]).	152
5.4	Variation of E-norm condition number $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ with the mean of $ \gamma(f) ^2$ across frequency bins $0 \leq f \leq L - 1$	175
5.5	Variation of steady-state normalized misalignment with E-norm condition number $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$.	175
5.6	Effect of tap selection on (a) \mathcal{M}_t and (b) \mathcal{M}_f for an adaptive filter length of $L = 256$. . .	180
5.7	Schematic of subselected tap-input vector and its frequency-domain quantity using 50% overlapping factor.	182

5.8	Input sequence partitioning for the j^{th} channel FLMS algorithm with arbitrary overlapping factor controlled by α	183
5.9	Normalized misalignment for WGN input with mean interchannel coherences of (a) 0.85, (b) 0.60 and (c) 0.53.	186
5.10	Normalized misalignment for speech input with mean interchannel coherences of (a) 0.85, (b) 0.60 and (c) 0.53.	186
5.11	Normalized misalignment using WGN input sequence for (a) NL-FLMS and (b) XMNL-FLMS [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-FLMS}} = 1$, $\mu_{\text{XMNL-FLMS}} = 0.4$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	188
5.12	Normalized misalignment using speech input sequence for (a) NL-FLMS and (b) XMNL-FLMS [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-FLMS}} = 1$, $\mu_{\text{XMNL-FLMS}} = 0.43$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	188
5.13	Normalized misalignment using WGN input sequence for (a) NL-FLMS α and (b) XMNL-FLMS α [$L_T = L_R = 800$, $L = 256$, $\alpha = 4$, $\mu_{\text{NL-FLMS}\alpha} = 1$, $\mu_{\text{XMNL-FLMS}\alpha} = 0.65$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	189
5.14	Normalized misalignment using speech input sequence for (a) NL-FLMS α and (b) XMNL-FLMS α [$L_T = L_R = 800$, $L = 256$, $\alpha = 4$, $\mu_{\text{NL-FLMS}\alpha} = 1$, $\mu_{\text{XMNL-FLMS}\alpha} = 0.65$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].	189
5.15	Effect of interchannel coherence on misalignment where interchannel coherence is controlled by addition of WGN to $x_2(n)$ giving interchannel coherences of (a) 0.90, (b) 0.67 and (c) 0.35 [$L_T = L_R = 1024$, $L = 1024$, $f_s = 16$ kHz, $\lambda = [1 - 1/(3L)]^L$, SNR = 30 dB].	195
6.1	Schematic of an in-car communication system.	205

List of Tables

2.1	Examples of computational complexity for MMax algorithms [$L = 1024, M = 512, A = 64, S = 128, K = 2$].	55
2.2	The Periodic-NLMS algorithm [19]	63
2.3	The Sequential-NLMS algorithm [19]	63
2.4	The SPU-NLMS algorithm [20]	64
2.5	The MMax-NLMS algorithm [21]	64
2.6	The MMax-AP algorithm	64
2.7	The MMax-RLS algorithm	65
3.1	Projection order and Γ_{n-k} for various algorithms	70
3.2	Steady-state misalignment for WGN input	90
4.1	The XMNL tap selection	141
4.2	The XMNL-NLMS algorithm	141
4.3	The XMNL-AP algorithm	142
4.4	The XMNL-RLS algorithm	142
5.1	Data sectioning matrices	196
5.2	The FLMS algorithm [22] [11]	197
5.3	The MDF algorithm [23] [11]	197
5.4	The two-channel frequency-domain adaptive algorithm [11]	198
5.5	The XM-FLMS algorithm	199
5.6	The XM-FLMS α algorithm	200

Abbreviations

AEC:	Acoustic echo cancellation
AP:	Affine projection algorithm
DTD:	Double-talk detector
ERLE:	Echo return loss enhancement
FIR:	Finite impulse response
FLMS:	fast-LMS algorithm
FLMSα:	fast-LMS algorithm with overlapping factor α
GMDFα:	Generalized multi-delay filtering with overlapping factor α
LEM:	Loudspeaker-enclosure-microphone
LMS:	Least-mean-square algorithm
MDF:	Multi-delay filtering structure
NLMS:	Normalized least-mean-square
PSTN:	Public switched telephone network
RLS:	Recursive least squares algorithm
SAEC:	Stereophonic acoustic echo cancellation
SNR:	Signal-to-noise ratio
WGN:	White Gaussian noise
WSS:	Wide-sense stationary
XM:	Exclusive-maximum tap selection

Notation

$[\cdot]_{m \times n}$	Matrix of dimension m rows \times n columns
$[\cdot]^T$	Matrix transpose operator
$[\cdot]$	Truncation operator
$*$	Complex conjugation
\otimes	Convolution operator
\odot	Element-by-element product (Schür product)
$\mathbf{0}_{m \times n}$	Null matrix of dimension m rows \times n columns
$\mathbf{I}_{m \times m}$	Identity matrix of dimension m rows \times m columns
$E\{\cdot\}$	Expectation operator
$\text{tr}\{\cdot\}$	Trace operator
$\ \cdot\ _2^2$	Squared l_2 -norm
n	Sample iteration
L	Length of adaptive filter
L_T	Length of transmission room impulse responses
L_R	Length of receiving room impulse responses
$s_T(n)$	Transmission room source
$s_R(n)$	Receiving room source
$y(n)$	Receiving room microphone signal
$\hat{y}(n)$	Filter output
$e(n)$	<i>a priori</i> error
$e_p(n)$	<i>a posteriori</i> error
$w(n)$	Uncorrelated measurement noise

$\mathbf{Q}(n)$	Diagonal tap selection control matrix
M	Number of taps selected for adaptation based on MMax tap selection criterion
$\mathcal{J}(n)$	Cost function
$\mathbf{g}_j(n)$	j^{th} channel far-end (transmission room) impulse response
$\mathbf{h}_j(n)$	j^{th} channel near-end (receiving room) impulse response
$\hat{\mathbf{h}}_j(n)$	j^{th} channel estimated response
$\mathbf{x}_j(n)$	j^{th} channel tap-input vector
$\tilde{\mathbf{x}}_j(n)$	j^{th} channel subselected tap-input vector given by $\mathbf{Q}_j(n)\mathbf{x}_j(n)$
$\mathbf{x}'_j(n)$	j^{th} channel non-linear processed input signal
λ	Forgetting factor with $0 << \lambda < 1$
$\bar{\lambda}_l$	l^{th} eigenvalue
$\mathbf{R}_{\mathbf{x}\mathbf{x}}$	Input autocorrelation matrix given by $E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$
$\Psi(n)$	Time-averaged input autocorrelation matrix given by $\sum_{i=1}^n \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i)$
$\Phi_{\mathbf{x}y}$	Cross-correlation vector between $\mathbf{x}(n)$ and $y(n)$ given by $E\{\mathbf{x}(n)y(n)\}$
$\Theta(n)$	Time-averaged cross-correlation between $\mathbf{x}(n)$ and $y(n)$ given by $\sum_{i=1}^n \lambda^{n-i}y(i)\mathbf{x}(i)$
$\mathbf{p}(n)$	Magnitude difference vector given by $ \mathbf{x}_1(n) - \mathbf{x}_2(n) $
\mathcal{M}	M-ratio defined as $\ \mathbf{Q}_n\mathbf{x}(n)\ _2^2 / \ \mathbf{x}(n)\ _2^2$
$\eta'(n)$	Misalignment defined as $\ \mathbf{h} - \hat{\mathbf{h}}(n)\ _2^2$
$\eta(n)$	Normalized misalignment defined as $\ \mathbf{h} - \hat{\mathbf{h}}(n)\ _2^2 / \ \mathbf{h}\ _2^2$
σ_x^2	Variance of zero mean input signal defined as $E\{x(n)x(n)\}$
m	Frequency-domain frame index
N	Block length of multi-delay filtering (MDF) structure
k	Block index for the MDF structure (Chapter 5)
\mathcal{K}	Total number of blocks for the MDF structure
ϑ_m	Magnitude weighting for exhaustive search technique in SAEC
ϑ_c	Coherence weighting for exhaustive search technique in SAEC defined as $1 - \vartheta_m$
ζ	Tap selection set for exhaustive search technique in SAEC
\mathcal{A}	Magnitude cost matrix
\mathcal{C}	Coherence cost matrix

\mathcal{V}	Total cost matrix for exhaustive search technique in SAEC
β	Non-linearity constant implemented using the half-wave rectifier for SAEC
δ	Regularization parameter
f_s	Sampling frequency
$\chi_E[\mathbf{A}]$	E-norm condition number of matrix \mathbf{A}
$\mathbf{F}_{2L \times 2L}$	Fourier matrix of dimension $2L \times 2L$
ξ	Control parameter for non-stationary unknown impulse response (Chapter 3)
$ \gamma(f) ^2$	Squared interchannel coherence
ψ	Tap-selection dependent variable defined as $L/(MM)$ (Chapter 3)
μ	Step-size parameter for NLMS, AP, FLMS based algorithms
μ_{mis}	Step-size with the lowest steady-state misalignment under non-stationary unknown system condition (Chapter 3)
\mathcal{B}	Number of partitioned blocks for SPU-NLMS
B	Number of blocks in SPU-NLMS selected for adaptation (Chapter 2)
$\mathbf{k}(n)$	Kalman gain for RLS
λ_c	Forgetting factor for computing $\ \mathbf{x}(n)\ _2^2$ using a recursive estimate (Chapter 2)
K	Affine projection order
\mathcal{N}	Periodic and Sequential-LMS control parameter
\mathcal{S}	Segmentation control parameter for Short-sort algorithm (Chapter 2)
\mathbf{s}_n	Noise process in modified first order Markov model (Chapter 3)
$\underline{\mathbf{S}}_{ju}$	Square matrix with elements containing cross-spectra between input channels j and u
\mathbf{v}_n	System mismatch vector defined as $\hat{\mathbf{h}}_n - \mathbf{h}_n$
Γ_{n-k}	Adaptation control matrix at sample iteration $n - k$ (Chapter 3)
ϕ, ϕ_a	Algorithmic dependent constant (see Table 3.2)
β_r	Tap-selection dependent variable in steady-state misalignment parameter for MMax-RLS (see Table 3.2)

- $\varphi(n)$ Scalar constant to illustrate non-unique solutions in SAEC
- ϵ Control parameter to vary interchannel coherence between two impulse responses in SAEC

Chapter 1

General Introduction

*Study the past if you
would define the future.*
Confucius (551-479 BC)

1.1 Developments in echo cancellation

ECHO is the repetition of sound caused by a delayed reflection of sound waves. In telecommunications networks, echoes prevent natural conversation when the speaker hears a delayed version of his utterance and since the human ear is sensitive to echo, even a round trip delay as short as tens of milliseconds (ms) inhibits natural conversation [1]. It should be noted however that not all echoes reduce voice quality. In order for phone conversations to sound natural, callers must be able to hear themselves speaking. For this reason, a short instantaneous echo known as the “side tone” is deliberately inserted, coupling the caller’s speech from the telephone mouthpiece to the earpiece so that the line sounds connected. However, longer round trip delays (exceeding 30 ms) can become annoying.

The telecommunications industry has sought means to reduce echo since the late 1950s with the advent of satellite communications where delays are considerably long. A telephone call connected via a geostationary satellite orbiting approximately 23,000 miles above Earth’s surface can experience a delay of approximately 500 to 600 ms [2]. Network

echo control in these early systems were first introduced in the form of echo-suppression devices for two- to four-wire hybrids [3] where an impedance mismatch causes received speech signal to be transmitted back to the source with a delay thus perceiving as echo. These echo suppressors were essentially voice-actuated switches which transmit signals and were subsequently turned off to suppress any returning echo signal. The main problem with these systems was that they only allow half-duplex communications. Network echo control underwent an evolutionary change in the early 1960s when echo cancellation theory was formally developed by AT&T Bell Labs [4] [5]. These systems utilized adaptive signal processing and reduced the echo by synthesizing an echo replica. Although a blockless echo suppressor [6] was proposed by generating the echo replica using an impulse generator, the adaptive echo canceller [1] [5] remains highly effective since the echo path is time-varying in practical implementations. Although experimental versions of echo cancellers were built and successfully tested, they were unfortunately too large and expensive for commercial service. With the advent of very-large-scale-integrated (VLSI) technology in the early 1970s, the first twelve-channel digital VLSI network echo canceller was implemented in 1978 [7] [8].

The challenge of providing hands-free telephone conversations has been recognized since the early 1970s [9]. With the development of hands-free tele-conferencing and in-car systems, acoustic coupling between the loudspeaker and microphone inhibits natural communication between users. Although many techniques were proposed including the use of frequency shifting, comb filters and center clipping to solve the acoustic echo cancellation (AEC) problem, one of the most efficient implementations was the extension of network echo cancellation adaptive algorithms [10]. Using the loudspeaker-enclosure-microphone (LEM) model [10], the concept of a feasible solution to the AEC problem employing an adaptive filtering algorithm is that if it is possible to provide a replica of the receiving room's impulse response, then decoupling of the loudspeaker and the microphone can be achieved.

Although direct application of network echo cancellation algorithms can be applied to AEC, the differences between the network and acoustic echo paths call for intensive

research for the latter application. The difficulties associated with the AEC problem result from two major factors: (i) the reverberation time of an office or a living room can be several hundred milliseconds long and (ii) the transfer characteristics of LEMs are sensitive to, for example, any movements of people or changes in the placement of furniture. The degree of acoustic echo cancellation depends on the closeness in approximation of the LEMs by its replica implemented using a linear filter whose primary objective is to model an impulse response dynamically. In a typical office or living room which exhibits reverberation time in the order of 50 to 300 ms [11], this translates to adapting 400 to 2400 filter coefficients at 8 kHz sampling frequency. In addition, room impulse responses are sensitive to movements of people and variations in temperature or pressure. A slight change of furniture setting, for example, can cause decoupling performance to degrade by approximately 15 dB for example cases shown in [12]. For this reason, adaptive filters are utilized to track and compensate any changes in the receiving room impulse responses.

The computational complexity of adaptive algorithms needs to be considered for efficient implementation. Assuming a sampling rate of 8 kHz and for a transversal filter length of, for example, 1024 coefficients (128 ms), approximately 8.192 million multiplications and the same number of additions per second are necessary to perform filtering. The need to reduce computational complexity is hence an important issue and as a consequence, a significant focus in AEC research has been to reduce the computational complexity of adaptive algorithms for applications requiring such high density or low cost. A result of this work is a class of partial update adaptive filtering algorithms that share the characteristic of executing tap update operations on only a subset of the filter coefficients at each iteration. With the reduction in complexity due to partial adaptation, it is normal to expect a degradation in performance of such algorithms. Hence, the challenge of researchers in this field is to develop tap selection schemes which reduces this degradation in performance.

More recently, modern applications such as multiparty room-to-room teleconferencing, multimedia desktop conferencing and interactive video online gaming call for more life-like multichannel sound transmission. One of the first two-channel hands-free stereo-

phonic conferencing system was implemented and tested in [13] where an echo suppressor, comprising of a comparator and logic controller, permits half-duplex stereophonic transmission. It was found that with stereophonic sound, speech intelligibility was enhanced and the ability for users to localize multiple far-end speakers was increased significantly hence reducing the “cocktail” party problem and enhancing tele-presence to users. With the introduction of such stereophonic systems, the need for stereophonic acoustic echo cancellation (SAEC) is inevitable. Although SAEC can be seen as a direct extension of the single channel AEC case, the SAEC problem is far more challenging as conventional single channel adaptive algorithms deployed for such an application suffer from non-unique solutions [14].

1.2 Research aim and thesis structure

The design of a hands-free system comprises issues such as adaptive step-size control, double talk detection and echo suppression [15] [16] [17]. Integrated systems such as the use of echo cancellation and noise reduction algorithms for AEC have also been considered in [16] [18]. The aim of this research work, however, is the development and analysis of adaptive algorithms employing tap selection for both single channel and stereophonic AEC in hands-free systems. As will be presented in this thesis, the concept of selective-tap algorithms is derived from that of partial update algorithms such that the former update all filter coefficients at each sample iteration although, as will be shown, their computational complexities are still lower than that of conventional (fully updating) adaptive algorithms.

This thesis is organized as follows: In Chapter 2, both the conventional and partial updating adaptive algorithms for the single channel AEC application are reviewed. Partial update algorithms built on the least-mean-square (LMS) and normalized-LMS (NLMS) algorithms such as the Periodic-LMS [19], Sequential-LMS [19], Selective-partial-updating NLMS (SPU-NLMS) [20] and MMax-NLMS [21] algorithms are reviewed and their performances are compared through simulation examples. The affine-projection (AP) and recursive least squares (RLS) algorithms employing MMax tap selection are developed with focus on the derivation of the MMax-RLS algorithm from a least-squares criterion.

The performances of the proposed MMax-AP and MMax-RLS are compared through simulation examples.

In Chapter 3, the steady-state misalignment of adaptive algorithms under time-varying unknown system conditions is analyzed. A general framework is developed such that the steady-state analysis can be applied to both the conventional (LMS, NLMS, AP and RLS) algorithms *and* their MMax variants. The aim of this analysis is to provide an insight of how the performances of such adaptive algorithms, in particular those employing MMax tap selection, are affected by conditions such as the degree of variation of the unknown system and the number of filter coefficients used for adaptation in single channel AEC. The proposed framework is presented and extensive formulations and discussions will be provided for each adaptive algorithm. Simulation results are provided to compare theoretical and experimental performances of each algorithm.

A novel application of selective-tap algorithms is the stereophonic acoustic echo cancellation (SAEC) problem as presented in Chapter 4. The main aim of employing selective-tap algorithms in SAEC is *not* to address the complexity reduction issue as for the single channel AEC case described in Chapters 2 and 3. Instead selective-tap algorithms are proposed for *reducing the interchannel coherence* so as to improve convergence performances of adaptive algorithms for SAEC. The motivation and aim of Chapter 4 is to develop a tap selection scheme for effective reduction in interchannel coherence whilst minimizing the degradation in convergence performance due to tap selection. As a proof of concept, an exhaustive search technique is presented to provide an insight of how tap selection can perform a reduction in interchannel coherence hence improving the rate of convergence of conventional adaptive algorithms for SAEC application. For effective implementation, an efficient tap selection algorithm known as the exclusive-maximum (XM) tap selection is proposed. As will be explained, the XM tap selection optimizes jointly the reduction in interchannel coherence and maximizing the “MMax-ness” of the selected taps so as to reduce the degradation in convergence performance of the proposed algorithms due to tap selection. Simulation results are presented to compare the improvement in convergence of the proposed algorithms over conventional algorithms without tap selection.

In Chapter 5, frequency-domain adaptive algorithms, in particular the fast-LMS (FLMS) algorithm [22] and the multi-delay filtering (MDF) structure [23] are reviewed. Following the approach as presented in [11], a general derivation of such adaptive algorithms in the frequency-domain is reviewed. The use of frequency-domain quantities allow the formulation of an explicit link between the interchannel coherence and the conditioning of the input autocorrelation matrix in SAEC. This relationship gives an insight of how interchannel coherence affects the conditioning of the input autocorrelation matrix which in turn degrades the misalignment performances of adaptive algorithms in SAEC. In addition, this relationship explains how interchannel coherence is reduced by the XM tap selection which consequently gives good convergence performances of the XM-based algorithms. Cases of tap selection, which can be achieved by subselecting tap-input vectors in time- or frequency-domain, are analyzed and their implications to the performances of frequency-domain algorithms employing tap selection are also discussed. Extensions of the XM tap selection to frequency-domain adaptive filtering are presented by considering both the 50% and an arbitrary overlapping factor between successive tap-input vectors. Simulation results are presented to verify theoretical analysis and to evaluate the performances of algorithms being developed.

1.3 Statement of originality, contributions and related publications

As far as the author is aware, the following aspects of this thesis are believed to be original and key contributions:

1. The extension of MMax tap selection [21] to the affine projection (AP) algorithm and the development of MMax-RLS from least-squares criterion as depicted in Chapter 2. The publication related to this contribution is [24].
2. A generalized framework for steady-state misalignment analysis which was proposed for a class of adaptive filtering algorithms (NLMS, AP and RLS) and their MMax-variants under both non-stationary and stationary unknown system conditions in

single channel AEC as discussed in Chapter 3. Publications related to this contribution are [24] [25].

3. The quantification of the closeness of a MMax subselected tap-input vector to that of a fully populated tap-input vector using the M-ratio measure \mathcal{M} . This ratio served as an optimization parameter for reducing the degradation in convergence performance due to tap selection in SAEC as discussed in Chapter 4. Key publications related to the M-ratio measure are [26] [27].
4. The development of a class of exclusive-maximum (XM) adaptive algorithms for SAEC application which, as a proof of concept, is derived from an exhaustive search technique, as depicted in Chapter 4. Publications related to this contribution are [28] [29] [30] [26] [27].
5. An analytical verification of how maximization of the M-ratio \mathcal{M} subjected to an exclusive tap selection constraint for both channels in SAEC is achieved by the XM tap selection is provided in Chapter 4. This verification has also been presented in [26].
6. Derivation of the link between interchannel coherence and the condition number of the two-channel input autocorrelation matrix as described in Chapter 5. This link allows one to explain the reduction in interchannel coherence and the improvement in conditioning of the input autocorrelation matrix due to XM tap selection in SAEC. This contribution has resulted in publications [31] [32].
7. Discussions on the subselection of tap-input vectors in time- and frequency-domain and their implications on performances of frequency-domain algorithms employing tap selection as presented in Chapter 5.
8. Development of frequency-domain XM algorithms for SAEC employing both the 50% and an arbitrary overlapping factor between successive tap-input vectors depicted in Chapter 5. The publication related to this contribution includes [32].

As this thesis is regarding the application of selective-tap algorithms for AEC application, the author has chosen not to include the following contribution in this thesis:

1. Development of the improved-proportionate multi-delay block adaptive filtering algorithm for network echo cancellation. This contribution has resulted in publication [33].
2. Development of adaptive algorithms for blind channel acoustic system identification based on work presented in [33].

Chapter 2

Algorithms Employing Tap Selection in Single Channel Acoustic Echo Cancellation

*The beginning of knowledge is the discovery
of something we do not understand.*

Frank Herbert (1920-1986)

2.1 Introduction

HANDS-FREE terminals have become increasingly popular due to the advent of video and desktop conferencing. The increase in popularity of in-car hands-free telephony due to the rising safety concern further calls for the need of hands-free systems. Whilst the introduction of hands-free telephony has brought about convenience and safety, the key issue of acoustic echo cancellation (AEC) needs to be addressed. In order for effective echo cancellation, a replica of the echo is generated by means of modelling the receiving room's impulse response using an adaptive filter. Implementation of an acoustic echo canceller poses great challenges due to (i) the long duration of the unknown echo path response, which can require several thousands of filter coefficients for accurate modelling, (ii) the highly time-varying nature of the echo response, and (iii) the need to train the

echo canceller using speech signal, which is coloured and statistically non-stationary.

As discussed in Section 1.1, a typical room impulse response in the region of 50 to 300 ms requires an adaptive filter of length 400 to 2400 at 8 kHz sampling frequency. The motivation for the introduction of selective-tap adaptive algorithms can be explained by considering the high computational load of adaptive algorithms. The normalized least-mean-square (NLMS) algorithm [34] [35] for an adaptive filter of length L requires approximately $2L$ multiply-accumulate (MAC) operations per sampling period of the signal. In the past, this rate of operation was considered high for typical telecommunications end-user equipment and researchers were therefore motivated to seek techniques that could reduce the computational complexity of adaptation without significantly degrading effectiveness in terms of its convergence rate or final misadjustment. More recently, the computational capability of low-cost processing hardware has increased very rapidly so that a typical NLMS implementation would not be seen as a heavy computational demand. However, new pressures on product design have emerged - the increase of user mobility imposes a requirement of low power consumption for portable battery powered equipment; the growth of telecommunications usage imposes a requirement of high density implementation for infrastructure equipment so that the number of simultaneous echo cancellers of given tap length that can be run within a specified MIP-budget (millions of instructions per second) is maximized. Both these requirements renew the motivation for low computational complexity, even with today's high speed processors. Consequently, significant focus for adaptive filter research in recent years has been to reduce the computational complexity of tap updates per iteration for applications requiring such high density or low cost.

Although an exhaustive review of complexity reduction techniques is beyond the scope of this chapter, several computational complexity reduction techniques have been identified. The use of post-filtering techniques is proposed as one of the methods to reduce the computational workload of processor chips. These techniques employ a conventional acoustic echo canceller of reduced length which models the direct path and the early reflections of the room impulse response, while the post-filter attenuates the residual echo corresponding to the late reverberation. In [36], the post-filtering is implemented using a

second adaptive filter and the combined system achieves a high echo attenuation even in the presence of high noise levels. It is found that this combined filtering approach requires approximately 1000 filter coefficients less in order to achieve comparable echo attenuation as that of a conventional echo canceller. The use of Wiener filtering and noise reduction techniques in place of an adaptive post-filter proposed in [37] further improve echo attenuation. It is noted that during double-talk, the attenuation performance increases with the post-filter length especially for lower frequencies (<1 kHz) where the echo and near-end input signal spectra are similar in terms of magnitude. The use of post-filtering and step-size control jointly has also been considered in [17]. The use of a frequency domain postfilter for background noise reduction and residual echo cancellation has been considered in [38].

Subband adaptive filtering (SAF) has been introduced in AEC to achieve complexity reduction whilst achieving an improved rate of convergence compared to full-band structures. In SAF, input signals are first partitioned into subbands and down-sampled using an analysis filter bank [39]. Consequently, adaptation with the down-sampled signals requires a lower complexity proportional to the down-sampling factor. Furthermore, in addition to the reduced spectral dynamic range, each subband may be adapted using different step-sizes matched to the energy of the input signal in that band hence achieving improved convergence [40]. The error signal of each subband is synthesized and up-sampled by the synthesis filter bank before being transmitted to the far-end. It should be noted that the gain in reducing processing power due to down-sampling far outweighs the overhead introduced by the analysis and synthesis filter banks. In [41] the authors formulated a modified subband structure where the error signals are computed in subbands while the adaptive filter coefficients are being updated in the full-band domain. Furthermore, utilizing the principle of minimal disturbance [35], the proposed algorithm achieves improved rate of convergence over the NLMS algorithm although a modest increase in computational complexity is required. To address the delay introduced by the analysis filter bank, several delayless SAF algorithms such as [42] [43] have been developed.

In recent years, partial update adaptive algorithms are proposed as an alternative form of complexity reduction of, in particular, the NLMS algorithm by updating only a subset of

filter coefficients at each sample iteration. These techniques allow implementation of single channel AEC with performance close to that of the conventional (fully updated) NLMS algorithm. One of the most recent tap selection schemes is the MMax tap selection [21] which, when applied to NLMS, is denoted as the MMax-NLMS algorithm. The aim of this chapter is to develop a class of MMax selective-tap algorithms for *single channel* AEC application. As will be seen in Chapter 4, these algorithms form the basis for the development of stereophonic acoustic echo cancellation (SAEC) selective-tap algorithms. This chapter is organized as follows: Section 2.2 presents a brief overview of the AEC problem which, in addition, introduces notations for use in this thesis. Two well-known performance measures for single channel AEC are explained and the explicit link between the two is shown in Section 2.3. Partial update adaptive algorithms and in particular the MMax-NLMS algorithm are reviewed in Section 2.5. A class of algorithms employing MMax tap selection is formulated by extending the MMax to the affine projection (AP) and recursive least squares (RLS) algorithms in Sections 2.6.1 and 2.6.2 respectively. The computational complexity of the proposed algorithms are discussed in Section 2.7 while simulation results comparing their performances are presented in Section 2.8.

2.2 The single channel acoustic echo cancellation problem

In hands-free systems, such as in-car telephony or tele-conferencing systems, the source of acoustic echo originates mainly from the acoustic coupling as well as possibly mechanical coupling between the microphone and loudspeaker. In this section, the single channel AEC problem is described and the NLMS algorithm is derived to address this problem.

2.2.1 Problem definition

Figure 2.1 shows a schematic diagram describing a typical single channel AEC system. A transmission room is depicted on the right where a microphone picks up time-varying signal $x(n)$ from a speech source via acoustic path

$$\mathbf{g}(n) = [g_0(n) \ g_1(n) \ \dots \ g_{L_T-1}(n)]^T, \quad (2.1)$$

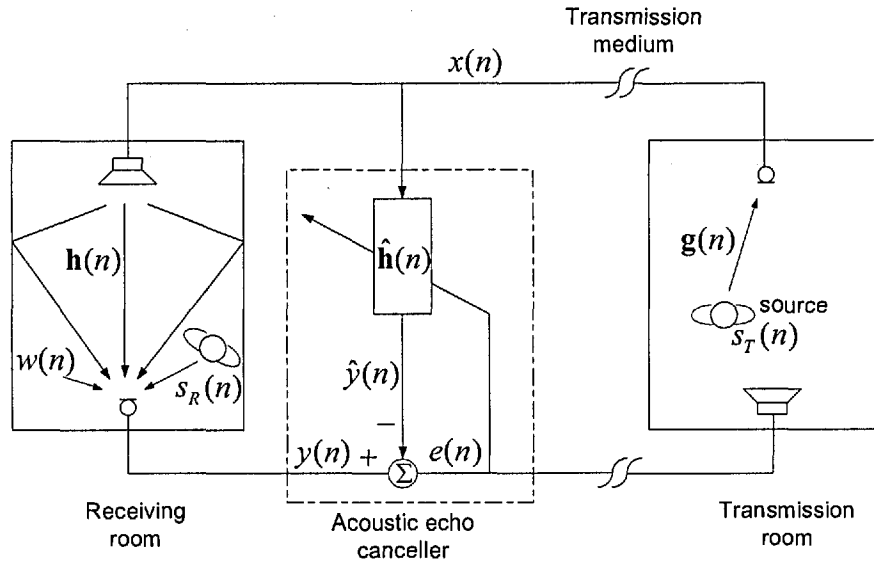


Figure 2.1: Schematic diagram of single channel acoustic echo cancellation.

which is the source to microphone impulse response in the transmission room and L_T is the length of $g(n)$ while the superscript T is the transposition operator. The input signal $x(n)$ is then transmitted to the loudspeaker in the receiving room depicted on the left which in turn is acoustically coupled to the receiving room's microphone via impulse response

$$\mathbf{h}(n) = [h_0(n) \ h_1(n) \ \dots \ h_{L_R-1}(n)]^T, \quad (2.2)$$

where L_R is the length of $\mathbf{h}(n)$. Defining uncorrelated background noise and near-end (receiving room) speech as $w(n)$ and $s_R(n)$ respectively, the received microphone signal $y(n)$ is then given by

$$y(n) = \mathbf{h}^T(n)\mathbf{x}(n) + w(n) + s_R(n), \quad (2.3)$$

where

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]^T \quad (2.4)$$

is the tap-input vector and L is the length of the adaptive filter. As will be explained in Section 2.2.2, the length of the adaptive filter is assumed to be the same as that of the unknown impulse response, i.e., $L = L_R$. The background noise $w(n)$ is assumed to be

zero mean and uncorrelated with $x(n)$.

If no echo canceller is present, the received signal $y(n)$, which contains a component of $x(n)$ given by (2.3), is transmitted back to the source with a delay and an attenuation, therefore impeding effective communication. Thus, an adaptive filter at the receiving end functions as an acoustic echo canceller by estimating receiving room's impulse response $\mathbf{h}(n)$ using filter coefficients $\hat{\mathbf{h}}(n)$ where

$$\hat{\mathbf{h}}(n) = [\hat{h}_0(n) \ \hat{h}_1(n) \ \dots \ \hat{h}_{L-1}(n)]^T. \quad (2.5)$$

The output of the echo canceller $\hat{y}(n)$ is subtracted from received signal $y(n)$ obtaining an *a posteriori* error signal $e_p(n)$ given by

$$\begin{aligned} e_p(n) &= y(n) - \hat{y}(n) \\ &= y(n) - \hat{\mathbf{h}}^T(n)\mathbf{x}(n) \\ &= [\mathbf{h}^T(n) - \hat{\mathbf{h}}^T(n)]\mathbf{x}(n) + w(n) + s_R(n). \end{aligned} \quad (2.6)$$

Thus for effective echo cancellation, the adaptive filter aims to model the receiving room's impulse response such that when $\hat{\mathbf{h}}(n) \approx \mathbf{h}(n)$, the component $\mathbf{x}(n)$ in $e_p(n)$ is significantly small. Note that the *a posteriori* error $e_p(n)$ as defined in (2.6) is computed after the adaptive filter coefficients have been updated. In contrast, the *a priori* error

$$e(n) = y(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \quad (2.7)$$

is computed using the previous impulse response estimate $\hat{\mathbf{h}}(n-1)$.

2.2.2 Assumptions

For simplicity and mathematical tractability, unless otherwise stated, the following are assumed in this thesis:

1. The length of the adaptive filter $\hat{\mathbf{h}}(n)$ is the same as that of the receiving room's impulse response $\mathbf{h}(n)$, i.e., $L = L_R$;

2. Receiving room source signal $s_R(n) = 0$ when input $x(n) \neq 0, \forall n$, i.e., no double-talk is present;
3. A finite impulse response (FIR) filter configuration is used.

Assumption (1) allows one to simplify mathematical derivations. In the realistic under-modelling case, where the length of the adaptive filter L is less than that of the receiving room's impulse response L_R , the adaptive filter must be of considerable length in order to achieve a good misalignment performance. As will be discussed in Section 2.7, the computational complexity of an adaptive algorithm increases monotonically with the length of the adaptive filter L . Consequently, current implementation considerations require a balance between the need for good misalignment performance and low computational complexity. In such practical implementations where $L < L_R$, the best achievable steady-state normalized misalignment in the absence of noise, is limited by the energies of the impulse response tail *not* modelled by the adaptive filter. This normalized misalignment can be expressed as [39] [44]

$$10 \log_{10} \left[\frac{\|\hat{\mathbf{h}}(n) - \mathbf{h}(n)\|_2^2}{\|\mathbf{h}(n)\|_2^2} \right] = 10 \log_{10} \left[\sum_{i=L}^{L_R-1} h_i^2(n) / \|\mathbf{h}(n)\|_2^2 \right] \text{ dB} , \quad (2.8)$$

where $\|\cdot\|_2^2$ is defined as the squared l_2 -norm operator and the estimated impulse response $\hat{\mathbf{h}}(n)$ is appended with $L_R - L$ zeros. Consequently, the performance in terms of normalized misalignment for an under-modelling case with $L < L_R$ will be lower than that of a perfect modelling case with $L = L_R$.

In most conversations, double-talk situations may arise when near-end speech signal $s_R(n) \neq 0$ at the receiving room while $x(n) \neq 0$. Under such situations, $s_R(n)$ may be perceived as a high level noise source which causes the adaptive filter to diverge and, as a result, annoying audible echo will be transmitted to the far-end source. This problem can be alleviated by employing a double-talk detector (DTD) such that once double-talk is detected, the adaptive filter coefficients are "frozen" and prohibited from adapting [45]. One of the earliest form of DTD algorithm for network echo cancellation is the Geigel algorithm where the adaptive filters are prohibited from adapting [7] if received signal

$y(n)$ is greater than half of the maximum element in $|\mathbf{x}(n)|$ where

$$|\mathbf{x}(n)| = [|x(n)| |x(n-1)| \dots |x(n-L+1)|]^T. \quad (2.9)$$

For AEC application however, a variable threshold has to be implemented due to the dynamic nature of the acoustic echo path [46]. Other DTD algorithms have been proposed for AEC including the cross-correlation [47] and the normalized cross-correlation methods [48]. Using an objective measure, it has been noted in [46] that for AEC application, the normalized cross-correlation method achieves the best performance compared to the Geigel algorithm and the cross-correlation technique. The performance recommendation of hands-free terminals in the presence of double talk is described in [49]. In this thesis, Assumption (2) is assumed and as such, the algorithm in study converges to its steady-state in the absence of double-talk.

Many physical systems can be well described by difference equations involving both the input and output. Hence linear time-invariant (LTI) infinite impulse response (IIR) models are commonly expected to possess better modelling capabilities than their finite impulse response (FIR) counterpart. However, it has been found that an IIR filter configuration does not show an advantage over an FIR configuration for AEC [44]. Due to the inherent stability of an FIR filter, the use of an FIR filter in Assumption (3) is assumed throughout this thesis.

2.3 Performance measure

2.3.1 Echo return loss enhancement

The echo return loss enhancement (ERLE) specified by the International Telecommunications Union (ITU) [50] measures the attenuation of the echo signals in an AEC system and is defined as

$$\text{ERLE} (n) = 10 \log_{10} \frac{y^2(n)}{e^2(n)} \text{ dB} . \quad (2.10)$$

It is usually applied to short frames of the signals and is often known as the segmented-ERLE. It can be seen that a higher ERLE corresponds to higher reduction in echo. The ITU-T G.167 recommendation for single channel acoustic echo controllers recommends an initial convergence rate of 20 dB per second [51].

2.3.2 Normalized misalignment

One of the most common performance measure is the mean-square deviation (MSD) [35] defined as $E\{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2\}$ where $E\{\cdot\}$ is the mathematical expectation operator. The instantaneous measure $\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2$ is commonly known as the misalignment such that when normalized with the energy of the unknown impulse response $\mathbf{h}(n)$, it is known as the normalized misalignment given by

$$\eta(n) = 10 \log_{10} \frac{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}(n)\|_2^2} \text{ dB} . \quad (2.11)$$

Hence the normalized misalignment measures the closeness of the estimated impulse response to that of the unknown impulse response and is particularly useful to study the tracking capability of adaptive algorithms. It should be noted that since the impulse response $\mathbf{h}(n)$ is unknown for practical systems, this measure is applicable only for synthetic simulations in which $\mathbf{h}(n)$ is known.

2.3.3 Relationship between normalized misalignment and error in single channel AEC

Assuming a time-invariant unknown impulse response \mathbf{h} , the normalized misalignment $\eta(n)$ can be expressed in terms of the *a priori* error $e(n)$ for a zero mean white Gaussian noise (WGN) input sequence by first defining the system mismatch vector

$$\mathbf{v}(n) = \hat{\mathbf{h}}(n) - \mathbf{h} \quad (2.12)$$

from which the squared *a priori* error, defined in (2.7) can be expressed as

$$e^2(n) = [w(n) - \mathbf{x}^T(n)\mathbf{v}(n-1)][w(n) - \mathbf{v}^T(n-1)\mathbf{x}(n)] , \quad (2.13)$$

where $w(n)$ is the uncorrelated noise as shown in Fig. 2.1. Defining $E\{\cdot\}$ as the mathematical expectation operator, it is noted that for large n , the system mismatch autocorrelation matrix

$$\begin{aligned} \mathbf{R}_v &= E\{\mathbf{v}(n)\mathbf{v}^T(n)\} \\ &\approx E\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\} , \end{aligned} \quad (2.14)$$

since after convergence, $\mathbf{v}(n) \approx \mathbf{v}(n-1)$. Letting the input autocorrelation matrix be

$$\begin{aligned} \mathbf{R}_{xx} &= E\{\mathbf{x}(n)\mathbf{x}^T(n)\} \\ &= \sigma_x^2 \mathbf{I}_{L \times L} \end{aligned} \quad (2.15)$$

where σ_x^2 and $\mathbf{I}_{L \times L}$ are the variance of $x(n)$ and the $L \times L$ identity matrix respectively, $E\{e^2(n)\}$ can then be expressed using (2.13) as

$$\begin{aligned} E\{e^2(n)\} &= E\{\text{tr}\{\mathbf{v}^T(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{v}(n-1)\}\} + E\{w^2(n)\} \\ E\{e^2(n)\} - E\{w^2(n)\} &= E\{\text{tr}\{\mathbf{v}^T(n-1)\mathbf{R}_{xx}\mathbf{v}(n-1)\}\} \\ &= E\{\text{tr}\{\mathbf{v}^T(n-1)\mathbf{R}_{xx}\mathbf{v}(n-1)\}\} \\ &= E\{\text{tr}\{\mathbf{v}(n-1)\mathbf{v}^T(n-1)\mathbf{R}_{xx}\}\} \\ &= \text{tr}\{\mathbf{R}_v\mathbf{R}_{xx}\} \\ &= \sigma_x^2 \text{tr}\{\mathbf{R}_v\} , \end{aligned} \quad (2.16)$$

where $\text{tr}\{\cdot\}$ is defined as the trace operator and the fourth equality follows from the trace identity $\text{tr}\{\mathbf{AB}\} = \text{tr}\{\mathbf{BA}\}$. Using (2.12) and (2.16), the normalized misalignment $\eta(n)$

can be expressed as

$$\begin{aligned}\eta(n) &= 10 \log_{10} \left[\frac{\text{tr}\{\mathbf{R}_v\}}{\|\mathbf{h}\|_2^2} \right] \\ &= 10 \log_{10} \left[\frac{E\{e^2(n)\} - E\{w^2(n)\}}{\sigma_x^2 \|\mathbf{h}\|_2^2} \right].\end{aligned}\quad (2.17)$$

It should be noted, from (2.3) and (2.7), that the *a priori* error $e(n)$ is lower bounded by the uncorrelated measurement noise $w(n)$. From (2.12) and (2.13), it can be seen that for small $e^2(n)$, $\hat{\mathbf{h}}(n) \rightarrow \mathbf{h}$ hence giving a low normalized misalignment $\eta(n)$ in (2.17) and a high ERLE in (2.10). It is important to note that for the stereophonic case, as will be discussed in Chapter 4, the condition $e^2(n) \rightarrow 0$ for a noiseless case $w(n) = 0$ does not necessarily imply $\hat{\mathbf{h}}(n) \rightarrow \mathbf{h}$. On the contrary, solutions for the estimated impulse response $\hat{\mathbf{h}}(n)$ under the condition $e^2(n) \rightarrow 0$ are non-unique [52] and depend on *both* the transmission and receiving rooms' impulse responses. Since the objective of the adaptive filter is to model the receiving room's impulse response \mathbf{h} , the normalized misalignment $\eta(n)$ is more applicable for single channel *and* stereophonic AEC and as a consequence, in a similar manner to published works, the normalized misalignment $\eta(n)$ is employed as the performance measure in this thesis.

2.4 The LMS and NLMS algorithms

The normalized least-mean-square (NLMS) algorithm [34] [35] [53] is an iterative formulation which solves the Wiener-Hopf equations recursively by employing the method of steepest descent. Exploiting the mean ergodic property [54] [55], filter coefficients are driven recursively such that as time progresses, they approach the optimal Wiener solution. In this section, the Wiener-Hopf equations are derived and how the method of steepest descent can be applied to form the NLMS algorithm for AEC application is discussed. As will be seen in Sections 2.5 and 5.3.3, this derivation will form the basis of selective-tap and frequency-domain adaptive filtering algorithms. Without the loss of generality, a noiseless case $w(n) = 0$ is assumed in this section. The case where $w(n) \neq 0$ will be considered in Chapter 3 for the purpose of algorithmic analysis under noisy conditions.

In addition, for mathematical tractability, input signal $x(n)$ is assumed to be zero mean white Gaussian noise (WGN) with variance σ_x^2 .

The *a posteriori* cost function \mathcal{J}_p is defined as the mean square error given by

$$\begin{aligned}\mathcal{J}_p &= E\{|e_p(n)|^2\} \\ &= E\{e_p^2(n)\},\end{aligned}\tag{2.18}$$

where the *a posteriori* error $e_p(n)$, as defined in (2.6), is a real quantity for AEC application. For the cost function to attain its minimum value, all elements of the gradient vector $\nabla\mathcal{J}_p$ must be simultaneously equal to zero,

$$\nabla_i\mathcal{J}_p = 0, \quad i = 0, 1, \dots, L-1,\tag{2.19}$$

where $\nabla_i\mathcal{J}_p = E\{\partial e_p^2(n)/\partial h_i(n)\}$. Defining $e^{\text{opt}}(n)$ as the error operating under the optimal condition $\nabla_i\mathcal{J}_p = 0$, it can be shown using (2.6) that

$$E\{e^{\text{opt}}(n)x(n-i)\} = 0,\tag{2.20}$$

for $i = 0, 1, \dots, L-1$. Equation (2.20) is known as the *principle of orthogonality* which states that at each sample iteration, the minimum error $e^{\text{opt}}(n)$ is orthogonal to each input sample [35]. To obtain the optimal solution $\hat{\mathbf{h}}^{\text{opt}}$ such that the cost function is minimized, (2.6) is substituted into (2.20) and using a temporary variable k , it can be shown that

$$\begin{aligned}E\left\{\left[y(n) - \sum_{k=0}^{L-1} \hat{h}_k^{\text{opt}} x(n-k)\right] x(n-i)\right\} &= 0 \\ \sum_{k=0}^{L-1} \hat{h}_k^{\text{opt}} E\{x(n-k)x(n-i)\} &= E\{x(n-i)y(n)\},\end{aligned}\tag{2.21}$$

for $i = 0, 1, \dots, L-1$. Having assumed $x(n)$ and $y(n)$ to be statistically invariant, the input autocorrelation matrix can be expressed as

$$\mathbf{R}_{\mathbf{xx}} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\},\tag{2.22}$$

while the cross-correlation between the tap-inputs of the filter and the desired response is expressed as

$$\Phi_{xy} = E\{\mathbf{x}(n)y(n)\} . \quad (2.23)$$

Using (2.21), the Wiener-Hopf equations [35] may be obtained as

$$\hat{\mathbf{h}}^{\text{opt}} = \mathbf{R}_{xx}^{-1} \Phi_{xy} . \quad (2.24)$$

The method of steepest descent [35] is a well-known optimization technique such that when applied to the Wiener filter, it allows *tracking* of time variations without having to invert \mathbf{R}_{xx} in (2.24). The basic concept of the method of steepest descent is that from an arbitrary starting point on the error performance surface, a small step is taken in the direction where the cost function decreases fastest. The filter coefficients thus progress towards the minimum point on the error performance surface as the number of iterations increases. For a simple illustrative case of $L = 2$, the error performance surface forms a paraboloid with a curvature determined by the eigenvalues of input autocorrelation matrix \mathbf{R}_{xx} . Letting μ be adaptation step-size, the recursive tap update equation is described by [34] [35],

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) - \mu \nabla \mathcal{J}(n) , \quad (2.25)$$

where

$$\mathcal{J}(n) = E\{e^2(n)\} \quad (2.26)$$

is the *a priori* error cost function and $e(n)$ is the *a priori* error defined in (2.7). The gradient $\nabla \mathcal{J}(n)$ can be simplified using (2.7) for $i = 0, 1, \dots, L-1$, giving

$$\begin{aligned} \nabla \mathcal{J}(n) &= \frac{\partial \mathcal{J}(n)}{\partial h_i(n)} \\ &= -2\Phi_{xy} + 2\mathbf{R}_{xx}\hat{\mathbf{h}}(n-1) . \end{aligned} \quad (2.27)$$

Substituting (2.27) into (2.25) and using (2.7), the recursive tap updating equation is given

by

$$\begin{aligned}
\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu[\Phi_{xy} - \mathbf{R}_{xx}\hat{\mathbf{h}}(n-1)] \\
&= \hat{\mathbf{h}}(n-1) + 2\mu\left[E\{\mathbf{x}(n)y(n)\} - E\{\mathbf{x}(n)\mathbf{x}^T(n)\}\hat{\mathbf{h}}(n-1)\right] \\
&\approx \hat{\mathbf{h}}(n-1) + 2\mu\mathbf{x}(n)\left[y(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n-1)\right] \\
&= \hat{\mathbf{h}}(n-1) + 2\mu\mathbf{x}(n)e(n) .
\end{aligned} \tag{2.28}$$

Note that, following the approach in [35], $E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$ and $E\{\mathbf{x}(n)y(n)\}$ are approximated by an instantaneous estimate. Defining $\bar{\lambda}_{\max}$ as the maximum eigenvalue of \mathbf{R}_{xx} , the adaptive step-size $0 < \mu < 1/\bar{\lambda}_{\max}$ serves as a control for adaptation speed [35]. It is further shown [56] [57] that under the condition $0 < \mu < 1/\text{tr}\{\mathbf{R}_{xx}\}$, the filter coefficients hover randomly about the Wiener solution. As will be shown through mathematical analysis in Chapter 3, a high value of μ will increase the rate of convergence but at the expense of steady-state misalignment. Equation (2.28) is also known as the least-mean-square (LMS) update equation¹ [59].

The normalized LMS (NLMS) algorithm is derived based on the principle of minimal disturbance [35] which minimizes the squared l_2 -norm of the change in filter coefficients from one iteration to the next given by

$$\|\hat{\mathbf{h}}(n) - \hat{\mathbf{h}}(n-1)\|_2^2, \tag{2.29}$$

subject to the constraint of

$$\hat{\mathbf{h}}^T(n)\mathbf{x}(n) = y(n) . \tag{2.30}$$

Applying the Lagrange multipliers and following similar approach to [35], the NLMS update equation is given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu \frac{\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{NLMS}}}, \tag{2.31}$$

where δ_{NLMS} is the regularization parameter which ensures stability during initialization

¹For readers' interests, Dr. B. Widrow's personal view on the discovery of the LMS algorithm can be found in [58].

when $\mathbf{x}(0) = \mathbf{0}_{L \times 1}$ is a null vector of dimension $L \times 1$.

2.5 Partial update adaptive algorithms

Adaptive filters with finite impulse response (FIR) are now widely used in many applications of signal processing in general and telecommunications in particular. The least-mean-square (LMS) algorithm and its normalized version (NLMS) [35] as described in Section 2.4 are the most common in practice because of their straightforward implementation and relatively low complexity compared to the better performing but substantially more complex least squares algorithms. The NLMS algorithm [35] requires approximately $2L$ multiply-accumulate (MAC) operations per sampling period of the signal and as discussed in Section 2.1, a significant focus in recent years has been to reduce the computational complexity of tap updates per iteration for applications requiring high density or low cost. A result of this work is a class of partial update adaptive filtering algorithms that share the characteristic of executing tap update operations on only a subset of the filter coefficients at each iteration. It is normal to expect that as the number of coefficients updated per iteration is reduced, the computational complexity is also reduced but at the expense of some loss in performance. Hence the goal of the designers of such partial update algorithms is to find ways to reduce the number of coefficients updated per iteration in a manner which degrades algorithm performance as little as possible. In this section, an overview of existing partial update algorithms is presented. As will be seen, these partial update algorithms can be broadly classified into (input) *data-independent* or *data-dependent* algorithms.

2.5.1 The Periodic-LMS and Sequential-LMS algorithms

The Periodic-LMS and Sequential-LMS algorithms [19] perform tap selection in a data-independent manner. In the Periodic-LMS algorithm, reduction in computation is achieved at each sample iteration n by updating filter coefficients periodically using the $\mathcal{N}[n/\mathcal{N}]^{\text{th}}$ instantaneous gradient estimate where $[\cdot]$ is defined as the truncation operator and $\mathcal{N} \in \{1, 2, \dots, L\}$. In addition, only taps satisfying the condition $(n + i) \bmod \mathcal{N} = 0$ for tap

indices $i = 0, 1, \dots, L - 1$ are updated at sample index n . Combining these two features and defining a $L \times L$ diagonal tap selection control matrix

$$\begin{aligned} \mathbf{Q}(n) &= \text{diag}\{\mathbf{q}(n)\} \\ &= \begin{bmatrix} q_0(n) & 0 & \cdots & 0 \\ 0 & q_1(n) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & q_{L-1}(n) \end{bmatrix}_{L \times L}, \end{aligned} \quad (2.32)$$

the Periodic-LMS update can be expressed as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu\mathbf{Q}(n)\mathbf{x}(l)e(l), \quad (2.33)$$

for $l = \mathcal{N}\lfloor n/\mathcal{N} \rfloor$. The tap selection elements for $i = 0, 1, \dots, L - 1$ are given as

$$q_i(n) = \begin{cases} 1, & \text{if } (n+i) \bmod \mathcal{N} = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (2.34)$$

while the *a priori* error $e(l)$ is expressed as

$$e(l) = y(l) - \mathbf{x}^T(l)\hat{\mathbf{h}}(l-1). \quad (2.35)$$

It can be seen that at each sample iteration, L/\mathcal{N} filter coefficients are updated such that after \mathcal{N} iterations all the filter coefficients have been updated once. For $\mathcal{N} = 1$, Periodic-LMS is equivalent to the LMS algorithm.

In contrast to Periodic-LMS, the Sequential-LMS algorithm [19] employs an instantaneous gradient estimate at each sample iteration for adaptation while only filter coefficients satisfying the condition $(n-i+1) \bmod \mathcal{N} = 0$ are updated. The Sequential-LMS update is expressed as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu\mathbf{Q}(n)\mathbf{x}(n)e(n), \quad (2.36)$$

where the diagonal tap selection control elements in $\mathbf{Q}(n)$ are now given, for $i =$

$0, 1, \dots, L - 1$, as

$$q_i(n) = \begin{cases} 1, & \text{if } (n - i + 1) \bmod \mathcal{N} = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2.37)$$

Similar to Periodic-LMS, for $\mathcal{N} = 1$, the Sequential-LMS algorithm reduces to the LMS algorithm. The computational complexity in terms of the number of multiplications required per sample iteration for the Periodic-LMS and Sequential-LMS algorithms are $(2L + 2)/\mathcal{N}$ and $L(1 + 1/\mathcal{N}) + 1$ respectively compared to $2L$ for the LMS algorithm.

Applying the principle of minimal disturbance [35] as described in Section 2.4, the Periodic-LMS and Sequential-LMS algorithms can be normalized following the same approach used in NLMS. Performance comparison for the resulting Periodic-NLMS and Sequential-NLMS algorithms will be described in Section 2.8. The Periodic-NLMS and Sequential-NLMS algorithms are summarized in Table 2.2 and Table 2.3 of Section 2.10.1 respectively.

2.5.2 The Selective-partial-update NLMS algorithm

As with the Periodic-LMS and Sequential-LMS algorithms discussed, the objective of Selective-partial-update NLMS (SPU-NLMS) [20] is to reduce computational complexity of the adaptive filter by updating only a subset of filter coefficients at each iteration. A key feature of SPU-NLMS is the partitioning of tap-input vector $\mathbf{x}(n) = [x(n) \ x(n - 1) \ \dots \ x(n - L + 1)]^T$ and the adaptive filter $\hat{\mathbf{h}}(n)$ into \mathcal{B} blocks of equal lengths hence giving

$$\mathbf{x}(n) = \left[\mathbf{x}_{s,1}^T(n) \ \mathbf{x}_{s,2}^T(n) \ \dots \ \mathbf{x}_{s,\mathcal{B}}^T(n) \right]^T, \quad (2.38)$$

$$\hat{\mathbf{h}}(n) = \left[\hat{\mathbf{h}}_{s,1}^T(n) \ \hat{\mathbf{h}}_{s,2}^T(n) \ \dots \ \hat{\mathbf{h}}_{s,\mathcal{B}}^T(n) \right]^T, \quad (2.39)$$

where the subscript s denotes for the SPU-NLMS algorithm. Defining δ_{SPU} as the regularization parameter, the block update

$$\hat{\mathbf{h}}_{s,i}(n) = \hat{\mathbf{h}}_{s,i}(n - 1) + 2\mu \frac{\mathbf{x}_{s,i}(n)e(n)}{\|\mathbf{x}_{s,i}(n)\|_2^2 + \delta_{\text{SPU}}} \quad (2.40)$$

for $i = 0, 1, \dots, \mathcal{B} - 1$, is derived as the solution to the constrained minimization problem [60]

$$\min_{1 \leq i \leq \mathcal{B}} \min_{\hat{\mathbf{h}}_{s,i}(n)} \left\| \hat{\mathbf{h}}_{s,i}(n) - \hat{\mathbf{h}}_{s,i}(n-1) \right\|_2^2, \quad (2.41)$$

subject to the constraint

$$\hat{\mathbf{h}}^T(n) \mathbf{x}(n) = y(n). \quad (2.42)$$

A decision can then be made at each iteration n on which B out of \mathcal{B} blocks to update. For $B = 1$, it is shown that the block, i , with the highest squared l_2 -norm of $\mathbf{x}_{s,j}(n)$ should be updated and this is found from the minimization

$$\begin{aligned} i &= \arg \min_{1 \leq j \leq \mathcal{B}} \left\| \hat{\mathbf{h}}_{s,j}(n) - \hat{\mathbf{h}}_{s,j}(n-1) \right\|_2^2 \\ &= \arg \min_{1 \leq j \leq \mathcal{B}} \left\| \frac{\mathbf{x}_{s,j}(n)e(n)}{\|\mathbf{x}_{s,j}(n)\|_2^2} \right\|_2^2 \\ &= \arg \min_{1 \leq j \leq \mathcal{B}} \left[\mathbf{x}_{s,j}(n) \left(\mathbf{x}_{s,j}^T(n) \mathbf{x}_{s,j}(n) \right)^{-1} \right]^T \left[\mathbf{x}_{s,j}(n) \left(\mathbf{x}_{s,j}^T(n) \mathbf{x}_{s,j}(n) \right)^{-1} \right] \\ &= \arg \min_{1 \leq j \leq \mathcal{B}} \left(\mathbf{x}_{s,j}^T(n) \mathbf{x}_{s,j}(n) \right)^{-1} \mathbf{x}_{s,j}^T(n) \mathbf{x}_{s,j}(n) \left(\mathbf{x}_{s,j}^T(n) \mathbf{x}_{s,j}(n) \right)^{-1} \\ &= \arg \min_{1 \leq j \leq \mathcal{B}} \frac{1}{\|\mathbf{x}_{s,j}(n)\|_2^2} \\ &= \arg \max_{1 \leq j \leq \mathcal{B}} \|\mathbf{x}_{s,j}(n)\|_2^2, \end{aligned} \quad (2.43)$$

where the second step arises from the constraint (2.42). As can be seen, the SPU-NLMS algorithm is a data-dependent partial update adaptive algorithm.

To update more than one block, $1 < B \leq \mathcal{B}$, the set $\mathcal{I}_B = \{i_1, i_2, \dots, i_B\}$ is defined to contain the indices of the blocks to be updated such that

$$\mathbf{x}_{s,\mathcal{I}_B}(n) = \left[\mathbf{x}_{s,i_1}^T(n) \ \mathbf{x}_{s,i_2}^T(n) \ \dots \ \mathbf{x}_{s,i_B}^T(n) \right]^T. \quad (2.44)$$

The SPU-NLMS update equation is then given as

$$\begin{aligned} \hat{\mathbf{h}}_{s,\mathcal{I}_B}(n) &= \hat{\mathbf{h}}_{s,\mathcal{I}_B}(n-1) + 2\mu \frac{\mathbf{x}_{s,\mathcal{I}_B}(n)e(n)}{\|\mathbf{x}_{s,\mathcal{I}_B}(n)\|_2^2 + \delta_{\text{SPU}}}, \\ \mathcal{I}_B &= \left\{ i \text{ for which } \|\mathbf{x}_{s,i}(n)\|_2^2 \text{ is one of the} \right. \\ &\quad \left. B \text{ greatest of } \|\mathbf{x}_{s,1}(n)\|_2^2, \dots, \|\mathbf{x}_{s,\mathcal{B}}(n)\|_2^2 \right\}. \end{aligned} \quad (2.45)$$

The SPU-NLMS algorithm is summarized in Table 2.4 of Section 2.10.1.

Extension of the selective-partial-update approach to include the affine projection adaptive algorithm is presented in [20]. Further discussion and analysis of the algorithm is also presented in [61]. It is noted that for large values of \mathcal{B} and small values of B , the SPU-NLMS algorithm may become unstable due to the high adaptive noise amplification brought about by the small value of $\|\mathbf{x}_{s,\mathcal{I}_B}(n)\|_2^2$. Consequently, bounds on the step-size μ are derived [61] for convergence in the mean squared sense and it is shown that an instantaneous estimate for μ giving the fastest convergence rate is $\mu = \|\mathbf{x}_{s,\mathcal{I}_B}(n)\|_2^2 / \|\mathbf{x}(n)\|_2^2$. This implies normalization by the l_2 -norm of the complete tap-input vector as in the MMax-NLMS algorithm. Such normalization has been employed for comparative simulations in Section 2.8. In addition, [61] employs the concept of set-membership adaptive filters [62] jointly with the partial updating scheme to obtain a set-membership partial update NLMS algorithm.

2.5.3 The Max-NLMS and MMax-NLMS algorithms

Based on [63], one of the earliest partial update algorithms is introduced in [64] where a family of NLMS algorithms is derived by minimizing the change in filter coefficients from one iteration to the next given by (2.29) using different l norms. By minimizing the l_1 -norm of filter coefficient change from sample iteration $n - 1$ to n , subject to the same constraint of (2.30), the adaptive algorithm degenerates to Max-NLMS [64] [65] where, being a data-dependent partial update algorithm, only one filter coefficient corresponding to the largest magnitude tap-input sample in $\mathbf{x}(n)$ is updated. For a specific set of input data given in [64], Max-NLMS outperforms the fully updated NLMS algorithm in terms of convergence rate. It can be seen that SPU-NLMS is equivalent to Max-NLMS when $\mathcal{B} = L$ and $B = 1$.

The single channel MMax-NLMS algorithm [66] is a direct extension of the Max-NLMS algorithm. The fundamental basis of MMax tap selection is that the sensitivity of the performance error to individual coefficient at each iteration depends on two factors namely (i) the shape of the mean-square error (MSE) surface and (ii) the location of

that coefficient at each time instance relative to the minimum of the MSE surface. This sensitivity is reflected in the steepness of the gradient vector components as described by (2.27). Using (2.28), the instantaneous gradient estimate in the direction of the i^{th} coefficient is $2x(n-i)e(n)$ where $i = 0, 1, \dots, L-1$ are the tap-indices of $\mathbf{x}(n)$ as shown in (2.4). Since all gradient components involve the quantity $2e(n)$, the MMax tap selection selects coefficients associated with the M largest values of $|x(n-i)|$ for updating. This can be interpreted as updating those coefficients contributing most to the trajectory of the adaptive algorithm towards the minimum point of the error performance surface. The MMax-NLMS algorithm can be expressed by first defining the $L \times L$ diagonal tap selection control matrix

$$\begin{aligned} \mathbf{Q}(n) &= \text{diag}\{\mathbf{q}(n)\} \\ &= \text{diag}\{q_0(n) \ q_1(n) \ \dots \ q_{L-1}(n)\}, \end{aligned} \quad (2.46)$$

where for tap-indices $i = 0, 1, \dots, L-1$,

$$q_i(n) = \begin{cases} 1, & |x(n-i)| \in \{M \text{ maxima of } |\mathbf{x}(n)|\}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.47)$$

while

$$|\mathbf{x}(n)| = [|x(n)| \ |x(n-1)| \ \dots \ |x(n-L+1)|]^T. \quad (2.48)$$

Consequently, the MMax-NLMS update equation is then given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu \frac{\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{NLMS}}}, \quad (2.49)$$

where as before, δ_{NLMS} and μ are the regularization parameter and step-size respectively.

For $M = 1$ and $M = L$, MMax-NLMS is equivalent to Max-NLMS [64] and NLMS respectively. The MMax-NLMS algorithm is summarized in Table 2.5 of Section 2.10.1.

2.6 Adaptive algorithms employing partial updates

Having reviewed partial update adaptive algorithms, the main contribution of this chapter is the extension of MMax tap selection to the affine projection (AP) and recursive least squares (RLS) algorithms. As will be seen through simulation examples in Section 2.8, the performance of MMax-NLMS, in terms of rate of convergence and steady-state normalized misalignment, is comparable to that of the NLMS algorithm for the case of $M = 0.5L$. Being a data-dependent algorithm, the MMax-NLMS outperforms the Periodic-NLMS, Sequential-NLMS and SPU-NLMS algorithms. As such, the MMax tap selection will be extended to the affine projection (AP) and recursive least squares (RLS) algorithms which will be denoted respectively as MMax-AP and MMax-RLS. The main benefit reported to motivate the introduction of AP and RLS selective-tap schemes is that they form the basis of selective-tap algorithms which are able to improve the conditioning of a two-channel autocorrelation matrix formed from correlated inputs such as occur in stereophonic acoustic echo cancellation (SAEC) [27], which will be presented in Chapter 4. In addition, as will be seen in this section, although the proposed MMax-AP and MMax-RLS algorithms employ MMax tap selection, they cannot be classified as partial update algorithms since, by virtue of their formulation, all coefficients are updated at each iteration. Consequently, the MMax-AP and MMax-RLS algorithms are classified as *selective-tap* algorithms. Nevertheless, as will be discussed in Section 2.7, the MMax-AP and MMax-RLS algorithms require less computation compared to the AP and RLS algorithms respectively.

2.6.1 The MMax affine projection algorithm

The affine projection (AP) algorithm [67] [68] incorporates multiple projections by concatenating past tap-input vectors from sample iteration n to $n - K + 1$ where K is defined as the projection order. In a similar manner, the approach for formulating the MMax-AP algorithm will be to concatenate the subselected tap-input vectors such that they propagate consistently from each sample iteration to the next. To formulate the MMax-AP algorithm [24], let

$$\tilde{\mathbf{x}}(n) = \mathbf{Q}(n)\mathbf{x}(n) \quad (2.50)$$

be the *subselected* tap-input vector where elements of the diagonal MMax tap selection matrix $\mathbf{Q}(n)$ are defined by (2.47). The concatenated subselected and full tap-input matrices of dimensions $K \times L$ are then defined as

$$\tilde{\mathbf{X}}_a(n) = [\tilde{\mathbf{x}}(n) \tilde{\mathbf{x}}(n-1) \dots \tilde{\mathbf{x}}(n-K+1)]^T, \quad (2.51)$$

$$\mathbf{X}_a(n) = [\mathbf{x}(n) \mathbf{x}(n-1) \dots \mathbf{x}(n-K+1)]^T \quad (2.52)$$

where the subscript a in $\tilde{\mathbf{X}}_a(n)$ and $\mathbf{X}_a(n)$ denotes for the AP algorithm. The tap update for the MMax-AP algorithm is then given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu \tilde{\mathbf{X}}_a^T(n) [\mathbf{X}_a(n) \mathbf{X}_a^T(n) + \delta_{AP} \mathbf{I}_{K \times K}]^{-1} \mathbf{e}(n), \quad (2.53)$$

where $\mathbf{I}_{K \times K}$ is the $K \times K$ identity matrix and

$$\mathbf{e}(n) = [e(n) e(n-1) \dots e(n-K+1)]^T \quad (2.54)$$

is the concatenated *a priori* error vector with each element computed using (2.7). Note that the update for MMax-AP in (2.53) normalizes with the full tap-input vector $\mathbf{X}_a(n)$ as oppose to $\tilde{\mathbf{X}}_a(n)$ since for small M , normalization with the latter can cause MMax-AP to become unstable. For projection order $K = 1$, MMax-AP is equivalent to MMax-NLMS. In addition, MMax-AP in general cannot be classified as a partial update algorithm since the tap update vector $\tilde{\mathbf{X}}_a^T(n) [\mathbf{X}_a(n) \mathbf{X}_a^T(n) + \delta_{AP} \mathbf{I}_{K \times K}]^{-1} \mathbf{e}(n)$ is fully populated and therefore every coefficient in $\hat{\mathbf{h}}(n)$ will be updated at each iteration. Consequently, MMax-AP is classified as a *selective-tap algorithm*. The MMax-AP algorithm is summarized in Table 2.6 of Section 2.10.1.

2.6.2 The MMax recursive least squares algorithm

One of the main disadvantages of the NLMS algorithm is the dependence of convergence rate on the eigenvalue spread of the input autocorrelation matrix $\mathbf{R}_{\mathbf{xx}}$ defined in (2.22). Specifically, input signals having a small eigenvalue spread exhibit higher rates of convergence compared to those having larger eigenvalue spread [35]. This affects the convergence

performance of NLMS for speech applications where the eigenvalue spread can be very significant (in order of several hundreds higher than for a WGN input). In contrast to the statistical approach discussed in Section 2.4, the method of least squares is a deterministic approach which involves the use of time-averages of $\mathbf{x}(n)$ and $y(n)$. In this section, the derivation of recursive least squares (RLS) algorithm employing MMax tap selection (MMax-RLS) will be presented.

The update equation of the RLS algorithm [35] is given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)e(n) , \quad (2.55)$$

where the $L \times 1$ vector $\mathbf{k}(n) = \Psi^{-1}(n)\mathbf{x}(n)$ is defined as the Kalman gain and

$$\Psi(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \quad (2.56)$$

is the $L \times L$ time-averaged autocorrelation matrix with forgetting factor $0 \ll \lambda < 1$. Direct extension of the MMax tap selection approach achieved by sorting the magnitude of $\mathbf{k}(n)$ in (2.55) will not give the desired convergence behavior especially for statistically non-stationary signals such as speech. This is because the Kalman gain depends on previous values of the time-averaged input autocorrelation matrix $\Psi(n)$ [27] given by

$$\mathbf{k}(n) = \Psi^{-1}(n)\mathbf{x}(n) , \quad (2.57)$$

where

$$\Psi^{-1}(n) = \frac{\lambda^{-1}\Psi^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n)} . \quad (2.58)$$

To address this, the tap-input vector $\mathbf{x}(n)$ is subsampled at each sample iteration based on the MMax tap selection criterion and $\Psi(n)$ is computed from the subselected tap-input vector $\tilde{\mathbf{x}}(n)$ giving $\tilde{\Psi}(n)$ where $\tilde{\mathbf{x}}(n)$ is defined in (2.50). This ensures that the subselected tap-input vectors propagate consistently through the memory of the RLS algorithm.

Similar to the normal equations in (2.24), the MMax-RLS algorithm [24] solves the

least-squares normal equations formed from $\tilde{\mathbf{x}}(n)$ given as

$$\hat{\mathbf{h}}(n) = \tilde{\Psi}^{-1}(n)\tilde{\Theta}(n) \quad (2.59)$$

where

$$\tilde{\Psi}(n) = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{x}}(i) \tilde{\mathbf{x}}^T(i), \quad (2.60)$$

$$\tilde{\Theta}(n) = \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{x}}(i) y(i), \quad (2.61)$$

and $y(i)$ is the receiving room's microphone signal at the i^{th} iteration as depicted in Fig. 2.1. The subselected time-averaged autocorrelation matrix $\tilde{\Psi}(n)$ can be expressed recursively as

$$\begin{aligned} \tilde{\Psi}(n) &= \tilde{\mathbf{X}}_r(n) \Lambda(n) \tilde{\mathbf{X}}_r^T(n) \\ &= \lambda \tilde{\Psi}(n-1) + \tilde{\mathbf{x}}(n) \tilde{\mathbf{x}}^T(n), \end{aligned} \quad (2.62)$$

where the subscript r in $\tilde{\mathbf{X}}_r(n)$ denotes for the MMax-RLS algorithm and $\tilde{\mathbf{X}}_r(n) = [\tilde{\mathbf{x}}(1) \tilde{\mathbf{x}}(2) \dots \tilde{\mathbf{x}}(n)]$ with $\Lambda(n) = \text{diag}\{\lambda^n \lambda^{n-1} \dots \lambda\}$. As before, the sub-selected tap-input vector is given as $\tilde{\mathbf{x}}(n) = \mathbf{Q}(n)\mathbf{x}(n)$ where elements of the MMax tap selection diagonal matrix $\mathbf{Q}(n)$ is defined in (2.47). In a similar manner, the time-averaged $L \times 1$ cross-correlation vector in (2.61) may be expressed recursively as

$$\begin{aligned} \tilde{\Theta}(n) &= \tilde{\mathbf{X}}_r(n) \Lambda(n) \mathbf{y}(n) \\ &= \lambda \tilde{\Theta}(n-1) + \tilde{\mathbf{x}}(n) y(n) \end{aligned} \quad (2.63)$$

where $\mathbf{y}(n) = [y(1) y(2) \dots y(n)]^T$.

Similar to the RLS algorithm, the MMax-RLS utilizes the matrix inversion lemma to compute $\tilde{\Psi}(n)$ efficiently. The matrix inversion lemma [35] [69] states that the inverse of $\mathbf{B} + \mathbf{b}\mathbf{b}^T$ is given by

$$(\mathbf{B} + \mathbf{b}\mathbf{b}^T)^{-1} = \mathbf{B}^{-1} - \frac{\mathbf{B}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{B}^{-1}}{1 + \mathbf{b}^T\mathbf{B}^{-1}\mathbf{b}}, \quad (2.64)$$

where \mathbf{B} and \mathbf{b} are of dimensions $L \times L$ and $L \times 1$ respectively. Letting $\mathbf{B} = \lambda \tilde{\Psi}(n-1)$ and $\mathbf{b} = \tilde{\mathbf{x}}(n)$, the inverse time-averaged input autocorrelation matrix $\tilde{\Psi}^{-1}(n)$ is expressed recursively as

$$\tilde{\Psi}^{-1}(n) = \frac{1}{\lambda} \left[\tilde{\Psi}^{-1}(n-1) - \tilde{\mathbf{k}}(n) \tilde{\mathbf{x}}^T(n) \tilde{\Psi}^{-1}(n-1) \right], \quad (2.65)$$

and the modified Kalman gain is then given by

$$\begin{aligned} \tilde{\mathbf{k}}(n) &= \frac{\lambda^{-1} \tilde{\Psi}^{-1}(n-1) \tilde{\mathbf{x}}(n)}{1 + \lambda^{-1} \tilde{\mathbf{x}}^T(n) \tilde{\Psi}^{-1}(n-1) \tilde{\mathbf{x}}(n)} \\ &= \lambda^{-1} \left[\tilde{\Psi}^{-1}(n-1) - \tilde{\mathbf{k}}(n) \tilde{\mathbf{x}}^T(n) \tilde{\Psi}^{-1}(n-1) \right] \tilde{\mathbf{x}}(n) \\ &= \tilde{\Psi}^{-1}(n) \tilde{\mathbf{x}}(n). \end{aligned} \quad (2.66)$$

The recursive solution to the normal equation given in (2.59), can be obtained by substituting the recursive form of $\tilde{\Theta}(n)$ and $\tilde{\Psi}^{-1}(n)$ in (2.63) and (2.65) into (2.59). Using (2.66), the MMax-RLS update equation is then expressed by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n) e(n), \quad (2.67)$$

where $e(n)$ is the *a priori* error as defined by (2.7).

Similar to the MMax-AP algorithm as described in Section 2.6.1, the MMax-RLS algorithm updates all the taps at each iteration since the modified Kalman gain vector $\tilde{\mathbf{k}}(n)$ is a fully populated column vector. Consequently, MMax-RLS is also considered as a *selective-tap* algorithm rather than a partial update algorithm. The MMax-RLS algorithm is depicted in Table 2.7 of Section 2.10.1.

2.7 Computational complexity

In this section, the computational complexity of algorithms employing MMax tap selection is examined. Although many factors contribute to the complexity of an algorithm, the relative complexity of the algorithms in terms of the total number of multiplications and comparisons per sample period is assessed here.

It should be noted that the computation of $\|\mathbf{x}(n)\|_2^2 = \mathbf{x}^T(n) \mathbf{x}(n)$ requires one multi-

plication using the recursive formulation [66]

$$\|\mathbf{x}(n)\|_2^2 = \|\mathbf{x}(n-1)\|_2^2 + x^2(n) - x^2(n-L) \quad (2.68)$$

or two multiplications using the recursive estimate

$$\|\mathbf{x}(n)\|_2^2 = \lambda_c \|\mathbf{x}(n-1)\|_2^2 + (1 - \lambda_c)x^2(n), \quad (2.69)$$

where $0 \ll \lambda_c < 1$ is the forgetting factor. The MMax tap selection requires a sorting operation to select the M largest tap-inputs at each iteration and can be achieved efficiently using for example the SORTLINE [70] or the Short-sort [71] routines. The Short-sort selects the largest A out of S elements from $[x(n) \ x(n-1) \ \dots \ x(n-S+1)]^T$ and then tracks them as they propagate through the memory of the filter with $S \ll L$ typically. The worst-case comparison load using Short-sort is $(1 + S - A)A/S$ comparisons per iteration compared to $2 + 2 \log_2 L$ used in the SORTLINE procedure [71]. Excluding the overhead of $\|\mathbf{x}(n)\|_2^2$ computation as described by either (2.68) or (2.69), the MMax-NLMS algorithm employing the SORTLINE procedure requires at most $L + M + 3 + 2 \log_2 L$ operations whereas $L + S + (1 + S - A)A/S$ operations are required for MMax-NLMS employing the Short-sort procedure (SM-NLMS).

The complexity of AP using the generalized Levinson algorithm is $2LK + 7K^2$ multiplies per sample period [45]. The MMax-AP algorithm employing the SORTLINE procedure requires an additional $2 + 2 \log_2 L$ sorting operations for the subselected tap-input vector $\tilde{\mathbf{x}}(n)$. However, due to a reduction in multiplications required when computing $\tilde{\mathbf{X}}_a^T(n) [\mathbf{X}_a(n) \mathbf{X}_a^T(n) + \delta_{AP} \mathbf{I}_{K \times K}]^{-1}$, the complexity for MMax-AP is $(M + L)K + 7K^2 + 2 + 2 \log_2 L$ operations per sample period [26].

The number of multiplications required for the RLS algorithm is $4L^2 + 3L + 2$ where an additional L multiplications are required for the tap updates. Due to the subselection of input vector $\tilde{\mathbf{x}}(n)$, the number of multiplications required for computing $\tilde{\Psi}(n)$ in MMax-RLS is $(M + L)L + 1$ while $L^2 + M$ multiplications are required for computing the Kalman gain. Hence the number of operations required for the MMax-RLS employing the

Table 2.1: Examples of computational complexity for MMax algorithms [$L = 1024, M = 512, A = 64, S = 128, K = 2$].

Algorithm	Sort Procedure	Multiplications and Comparisons	Examples
SM-NLMS	Short-sort	$L + S + (1 + S - A)A/S$	1.18×10^3
MMax-NLMS	SORTLINE	$L + M + 3 + 2 \log_2 L$	1.56×10^3
NLMS	-	$2L$	2.05×10^3
MMax-AP	SORTLINE	$(M + L)K + 7K^2 + 2 + 2 \log_2 L$	3.12×10^3
AP	-	$2LK + 7K^2$	4.12×10^3
MMax-RLS	SORTLINE	$L(L + 3M + 2) + M + 3 + 2 \log_2 L$	2.62×10^6
RLS	-	$4L^2 + 3L + 2$	4.20×10^6

SORTLINE procedure is at most $L(L + 3M + 2) + M + 3 + 2 \log_2 L$ per sample period [26].

As an illustrative example, an acoustic impulse response of 128 milliseconds (ms) at 8 kHz sampling frequency corresponds to $L = 1024$ and for an arbitrarily chosen $M = 512$, the number of operations required by MMax-NLMS, MMax-AP and MMax-RLS employing the SORTLINE algorithm is approximately 76.0%, 75.7% and 62.5% of the number for NLMS, AP and RLS respectively. Hence, although the MMax-AP and MMax-RLS algorithms update all coefficients at each sample iteration, their computation is nevertheless less than AP and RLS respectively. The computational complexity for the algorithms described are summarized in Table 2.1 with the number of multiplications and sorting operations computed for an example case of $L = 1024, M = 512, K = 2, A = 64$ and $S = 128$.

2.8 Simulation results

2.8.1 Experimental setup

Comparative results for the partial update and selective-tap algorithms as described in Sections 2.5 and 2.6 are presented in this section. For all simulations, impulse responses

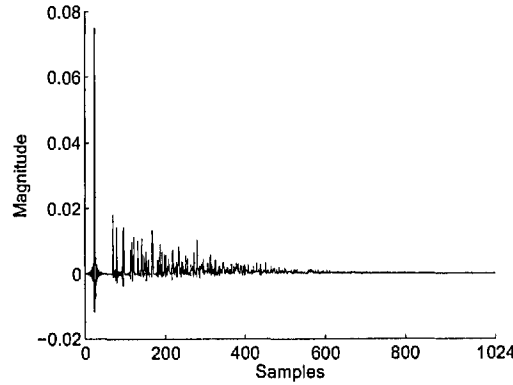


Figure 2.2: Receiving room impulse response $h(n)$ generated using the method of images at $f_s = 8$ kHz and $L_R = 1024$.

of the transmission and receiving rooms are given respectively as

$$\mathbf{g}(n) = [g_0(n) \ g_1(n) \ \dots \ g_{L_T-1}(n)]^T, \quad (2.70)$$

$$\mathbf{h}(n) = [h_0(n) \ h_1(n) \ \dots \ h_{L_R-1}(n)]^T, \quad (2.71)$$

and are generated using the method of images [72] with L_T and L_R being the lengths of the transmission and receiving rooms' impulse responses respectively. One microphone is placed in the centre of each room of dimension $3 \times 4 \times 5$ m. The source is then placed 1 m in-front-of the microphone in the transmission room. In a similar manner, impulse response $h(n)$ is generated with the receiving room microphone positioned 1.1 m in-front-of the loudspeaker. Figure 2.2 shows an example of the acoustic impulse response $h(n)$ generated at $f_s = 8$ kHz sampling frequency using the method of images with $L_R = 1024$. With reference to Fig. 2.1, tap-input vector

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]^T \quad (2.72)$$

is generated by convolving a source (WGN or speech) with $\mathbf{g}(n)$. In order to reflect realistic application, the undermodelling case of $L < L_R$ is used for all experiments. Defining \otimes as the convolution operator, the received signal $y(n)$ as defined in (2.3), is generated by $h(n) \otimes x(n)$ and an uncorrelated WGN $w(n)$ with zero mean is added such that an SNR as depicted in each experiment is achieved.

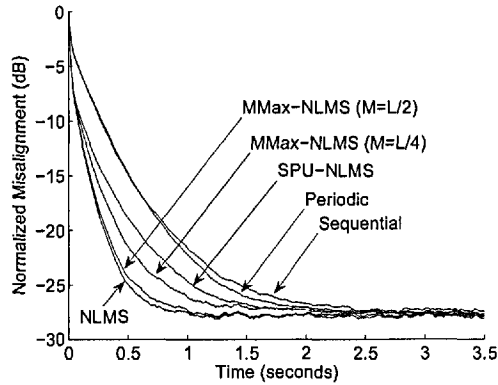


Figure 2.3: Normalized misalignment comparison for single channel partial update algorithms using WGN input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mathcal{N} = 2$, $B = 16$, $B = 16$, $\mu_{\text{NLMS}} = 0.7$, $\mu_{\text{Periodic}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, $\mu_{\text{SPU}} = 0.6$, $\mu_{\text{Sequential}} = 0.5$, SNR = 30 dB].

2.8.2 NLMS-based simulations

The convergence performance of the fully updated NLMS algorithm is compared to the Periodic-NLMS, Sequential-NLMS, SPU-NLMS and MMax-NLMS algorithms in Fig. 2.3. The impulse responses $\mathbf{g}(n)$ and $\mathbf{h}(n)$ are each of length $L_T = 1024$ and $L_R = 1024$ respectively. A sampling frequency of $f_s = 8$ kHz is used in this simulation while an adaptive filter of length $L = 512$ is chosen such that the adaptive filter undermodels the unknown system. The normalized misalignment is defined in (2.11) and is reproduced here for convenience

$$\eta(n) = 10 \log_{10} \frac{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}(n)\|_2^2} \text{ dB} . \quad (2.73)$$

Figure 2.3 shows the averaged normalized misalignment plot of 5 independent trials for each of the above mentioned algorithms using a WGN source sequence with zero mean and unit variance. The MMax-NLMS algorithm is tested with $M = L/2$ and $M = L/4$. For both Periodic-NLMS and Sequential-NLMS, $\mathcal{N} = 2$ is used, while for SPU-NLMS, $B = 16$ out of $\mathcal{B} = 32$ blocks are updated so that $L/2$ coefficients are updated at each iteration. The step-size of each algorithm is chosen experimentally so that all algorithms achieve the same asymptotic performance in terms of steady-state normalized misalignment which then allows one to compare their relative rate of convergence. This corresponds to $\mu_{\text{NLMS}} = 0.7$ for NLMS, $\mu_{\text{Periodic}} = 0.7$ for Periodic-NLMS, $\mu_{\text{MMax}} = 0.7$ for MMax-NLMS, $\mu_{\text{SPU}} = 0.6$

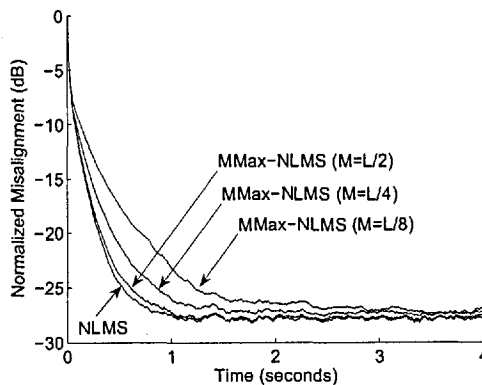


Figure 2.4: Normalized misalignment for single channel NLMS and MMax-NLMS using WGN input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{\text{NLMS}} = 0.7$, $\mu_{\text{MMax}} = 0.7$, SNR = 30 dB].

for SPU-NLMS and $\mu_{\text{Sequential}} = 0.5$ for Sequential-NLMS. An uncorrelated zero mean WGN sequence $w(n)$ is added to achieve an SNR of 30 dB in this simulation example.

Note that for full adaptation $M = L$, MMax-NLMS is equivalent to NLMS. It can be seen that NLMS achieves the highest rate of convergence since all taps are adapted at each sample iteration. Being data-dependent, the MMax-NLMS and SPU-NLMS algorithms outperform the data-independent Periodic-NLMS and Sequential-NLMS algorithms. For the case of MMax-NLMS with $M = 0.5L$, the convergence is close to that of NLMS suffering less than 1 dB degradation in normalized misalignment during convergence. For this experimental setup, it has been found that the NLMS algorithm achieves an ERLE, defined by (2.10), of 20 dB in approximately 0.25 s. Figure 2.4 shows additional results for MMax-NLMS using the same experimental setup as before. It can be seen that the rate of convergence reduces gracefully with M while approximately the same steady-state normalized misalignment is reached for each case of M .

The variation of misalignment with tap selection size M for MMax-NLMS using speech signal from a male talker is shown in Fig. 2.5 with $M = L/2$ and $M = L/4$. The step-sizes for MMax-NLMS and NLMS are $\mu_{\text{MMax}} = 0.7$ and $\mu_{\text{NLMS}} = 0.7$ respectively while $f_s = 8$ kHz, $L_T = L_R = 1024$, $L = 512$ are used. As before, an uncorrelated zero mean WGN $w(n)$ is added to achieve an SNR of 30 dB in this simulation example where the SNR is computed using the whole utterance of the speech sequence. A graceful

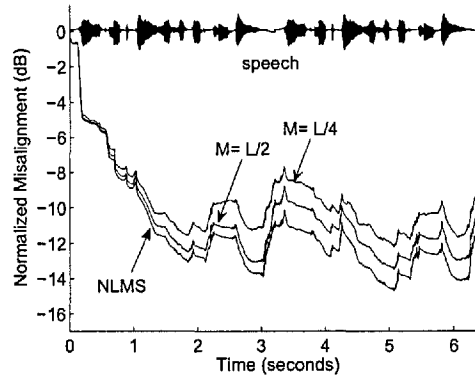


Figure 2.5: Normalized misalignment for single channel NLMS and MMax-NLMS using speech input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{NLMS} = 0.7$, $\mu_{MMax} = 0.7$, SNR = 30 dB].

degradation in convergence performance for reducing M can be observed in this speech signal example. In addition, the performance of MMax-NLMS for $M = L/2$ is close to that of NLMS, suffering approximately 1 to 2 dB degradation in normalized misalignment for this simulation example.

2.8.3 AP-based simulations

The effect of MMax tap selection on the affine projection algorithm is studied for the case of $M = L/2$, $L/4$ and $L/8$ with $L = 512$ using a WGN input sequence with zero mean and unit variance. As before, the sampling frequency for this simulation is $f_s = 8$ kHz while impulse responses $g(n)$ and $h(n)$ are each of length $L_T = 1024$ and $L_R = 1024$ respectively. An SNR of 30 dB is achieved using an additive WGN with zero mean while the affine projection order of $K = 2$ is used. For each case of M , the normalized misalignment is averaged over 5 independent trials and plotted as shown in Fig. 2.6. Similar to MMax-NLMS, the rate of convergence reduces gracefully with the number of taps being updated M for each iteration while the performance of MMax-AP (in terms of both the rate of convergence and steady-state misalignment) is close to that of AP for $M = L/2$.

The variation of misalignment with tap selection for MMax-AP using speech signal from a male talker is shown in Fig. 2.7 with $M = L/2$ and $M = L/4$. The step-sizes for MMax-AP and AP are $\mu_{MMax} = 0.7$ and $\mu_{AP} = 0.7$ respectively while $f_s = 8$ kHz,

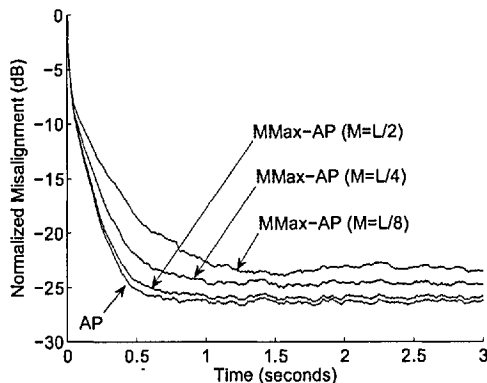


Figure 2.6: Normalized misalignment for single channel AP and MMax-AP using WGN input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu = 0.7$, $K = 2$, SNR = 30 dB].

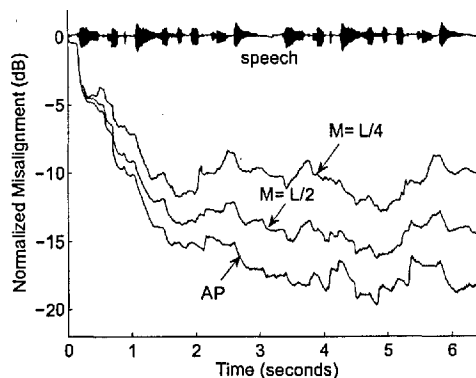


Figure 2.7: Normalized misalignment for single channel AP and MMax-AP using speech input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\mu_{AP} = 0.7$, $\mu_{MMax} = 0.7$, SNR = 30 dB].

$L_T = L_R = 1024$, $L = 512$ and SNR= 30 dB are used. As before, the SNR is computed using the whole utterance of the speech sequence. A graceful degradation in convergence performance can be seen when M is reduced for this speech signal example. In addition, the performance of MMax-AP for $M = L/2$ is close to that of AP such that approximately 2 dB degradation in normalized misalignment is observed during convergence in this simulation example.

2.8.4 RLS-based simulations

The effect of MMax tap selection on the RLS algorithm is shown in Fig. 2.8 using a WGN source sequence with zero mean and unit variance. In this simulation example, $L_T = L_R =$

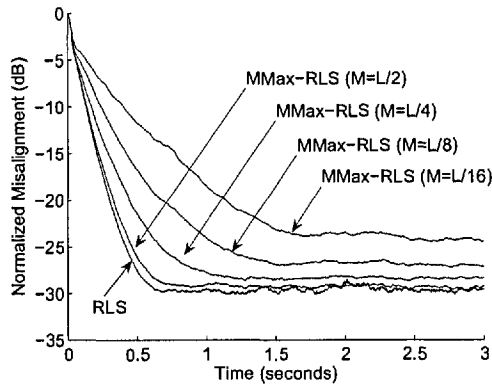


Figure 2.8: Normalized misalignment for single channel RLS and MMax-RLS [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\lambda = 0.9993$, SNR = 30 dB].

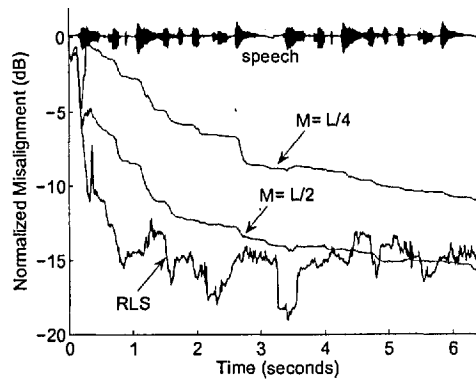


Figure 2.9: Normalized misalignment for single channel RLS and MMax-RLS using speech input [$L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz, $\lambda = 0.9998$, SNR = 30 dB].

1024, $L = 512$, $f_s = 8$ kHz and SNR= 30 dB. A forgetting factor of $\lambda = 1 - 1/(3L) = 0.9993$ is used [39]. The normalized misalignment is averaged over 5 independent trials for each case of $M = L/2, L/4, L/8$ and $L/16$. As before, the rate of convergence can be seen to reduce gracefully with M . In addition, the performance of MMax-RLS in terms of steady-state normalized misalignment degrades with reducing M such that compared to the fully updated RLS algorithm, an approximate degradation of 5 dB in normalized misalignment is observed for $M = L/16 = 32$.

The effect of MMax tap selection on the RLS algorithm for a speech input sequence is shown in Fig. 2.9. A forgetting factor of $\lambda = 1 - 1/(10L) = 0.9998$ is used [39] with $L_T = L_R = 1024$, $L = 512$, $f_s = 8$ kHz and SNR= 30 dB where the SNR is computed using

the whole utterance of the speech sequence. Similar to the MMax-NLMS and MMax-AP algorithms, the rate of convergence of the RLS algorithm degrades with reducing M . More importantly, the performance of MMax-RLS in terms of convergence rate and steady-state normalized misalignment for the case of $M = L/2$ is close to that of the RLS algorithm compared to $M = L/4$.

2.9 Conclusions

In this chapter, a brief overview of the single channel AEC problem has been presented and several partial update adaptive algorithms including the Periodic-LMS, Sequential-LMS, SPU-NLMS and MMax-NLMS algorithms have been reviewed. It has been shown through simulation examples that, among the partial update algorithms considered, the MMax-NLMS algorithm achieves the fastest rate of convergence and hence, the main contribution of this chapter is the formulation of AP and the derivation of the RLS algorithm employing MMax tap selection giving MMax-AP and MMax-RLS respectively. Comparative simulation results showed that convergence rates of the MMax-based algorithms are comparable to that of their corresponding fully updated algorithms for the case of $M = 0.5L$. The variation of convergence rate with M has also been presented for the case of MMax-AP and MMax-RLS showing the graceful degradation in performance for reducing M . The degradation in steady-state normalized misalignment performance for the MMax-based algorithms are insignificant for the cases studied here when $M = 0.5L$. This modest degradation in steady-state misalignment will be analyzed mathematically in the context of time-varying unknown system identification in Chapter 3. In addition, the robustness of NLMS, AP and RLS to MMax tap selection for $M = 0.5L$ will be exploited for stereophonic acoustic echo cancellation (SAEC) in Chapters 4 and 5.

2.10 Appendix

2.10.1 Partial updating and selective-tap algorithms

Table 2.2: The Periodic-NLMS algorithm [19]

$$\begin{aligned}
 l &= \mathcal{N}\lfloor n/\mathcal{N} \rfloor \\
 \hat{y}(n) &= \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \\
 e(l) &= y(l) - \hat{y}(l) \\
 \mathbf{Q}(n) &= \text{diag}\{q_0(n) \ q_1(n) \ \dots \ q_{L-1}(n)\} \\
 \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu \frac{\mathbf{Q}(n)\mathbf{x}(l)e(l)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{Periodic}}} \\
 q_i(n) &= \begin{cases} 1, & \text{if } (n+i) \bmod \mathcal{N} = 0 \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}$$

Table 2.3: The Sequential-NLMS algorithm [19]

$$\begin{aligned}
 \hat{y}(n) &= \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \\
 e(n) &= y(n) - \hat{y}(n) \\
 \mathbf{Q}(n) &= \text{diag}\{q_0(n) \ q_1(n) \ \dots \ q_{L-1}(n)\} \\
 \hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu \frac{\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{Sequential}}} \\
 q_i(n) &= \begin{cases} 1, & \text{if } (n-i+1) \bmod \mathcal{N} = 0 \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}$$

Table 2.4: The SPU-NLMS algorithm [20]

$$\begin{aligned}
\hat{y}(n) &= \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \\
e(n) &= y(n) - \hat{y}(n) \\
\hat{\mathbf{h}}(n) &= [\hat{\mathbf{h}}_{s,1}^T(n) \hat{\mathbf{h}}_{s,2}^T(n) \dots \hat{\mathbf{h}}_{s,B}^T(n)]^T \\
\mathbf{x}(n) &= [\mathbf{x}_{s,1}^T(n) \mathbf{x}_{s,2}^T(n) \dots \mathbf{x}_{s,B}^T(n)]^T \\
\hat{\mathbf{h}}_{s,\mathcal{I}_B}(n) &= \hat{\mathbf{h}}_{s,\mathcal{I}_B}(n-1) + 2\mu \frac{\mathbf{x}_{s,\mathcal{I}_B}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{SPU}}} \\
\mathcal{I}_B &= \{ i, \text{ if } \|\mathbf{x}_{s,i}(n)\|_2^2 \in \text{B greatest of } \|\mathbf{x}_{s,1}(n)\|_2^2, \dots, \|\mathbf{x}_{s,B}(n)\|_2^2 \}
\end{aligned}$$

Table 2.5: The MMax-NLMS algorithm [21]

$$\begin{aligned}
\hat{y}(n) &= \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \\
e(n) &= y(n) - \hat{y}(n) \\
\mathbf{Q}(n) &= \text{diag}\{q_0(n) q_1(n) \dots q_{L-1}(n)\} \\
\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu \frac{\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{NLMS}}} \\
q_i(n) &= \begin{cases} 1, & \text{if } |x(n-i)| \in \{M \text{ maxima of } |\mathbf{x}(n)|\} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

Table 2.6: The MMax-AP algorithm

$$\begin{aligned}
\mathbf{X}_a(n) &= [\mathbf{x}(n) \mathbf{x}(n-1) \dots \mathbf{x}(n-K+1)]^T \\
\mathbf{Q}(n) &= \text{diag}\{q_0(n) q_1(n) \dots q_{L-1}(n)\} \\
\tilde{\mathbf{x}}(n) &= \mathbf{Q}(n)\mathbf{x}(n) \\
\tilde{\mathbf{X}}_a(n) &= [\tilde{\mathbf{x}}(n) \tilde{\mathbf{x}}(n-1) \dots \tilde{\mathbf{x}}(n-K+1)]^T \\
\mathbf{y}(n) &= [y(n) y(n-1) \dots y(n-K+1)]^T \\
\hat{\mathbf{y}}(n) &= \mathbf{X}_a(n)\hat{\mathbf{h}}(n-1) \\
\mathbf{e}(n) &= \mathbf{y}(n) - \hat{\mathbf{y}}(n) \\
\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu \tilde{\mathbf{X}}_a^T(n) [\mathbf{X}_a(n)\mathbf{X}_a^T(n) + \delta_{\text{AP}}\mathbf{I}_{K \times K}]^{-1} \mathbf{e}(n) \\
q_i(n) &= \begin{cases} 1, & \text{if } |x(n-i)| \in \{M \text{ maxima of } |\mathbf{x}(n)|\} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

Table 2.7: The MMax-RLS algorithm

$\mathbf{Q}(n)$	$= \text{diag}\{q_0(n) \ q_1(n) \ \dots \ q_{L-1}(n)\}$
$\tilde{\mathbf{x}}(n)$	$= \mathbf{Q}(n)\mathbf{x}(n)$
$\tilde{\mathbf{k}}(n)$	$= \frac{\tilde{\Psi}^{-1}(n-1)\tilde{\mathbf{x}}(n)}{\lambda + \tilde{\mathbf{x}}^T(n)\tilde{\Psi}^{-1}(n-1)\tilde{\mathbf{x}}(n)}$
$\hat{y}(n)$	$= \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$
$e(n)$	$= y(n) - \hat{y}(n)$
$\hat{\mathbf{h}}(n)$	$= \hat{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n)e(n)$
$\tilde{\Psi}^{-1}(n)$	$= \frac{1}{\lambda} [\tilde{\Psi}^{-1}(n-1) - \tilde{\mathbf{k}}(n)\tilde{\mathbf{x}}^T(n)\tilde{\Psi}^{-1}(n-1)]$
$q_i(n)$	$= \begin{cases} 1, & \text{if } x(n-i) \in \{M \text{ maxima of } \mathbf{x}(n) \} \\ 0, & \text{otherwise} \end{cases}$

Chapter 3

Tracking Performance of MMax Algorithms Under Time-varying Unknown System Conditions

Arithmetic is where the answer is right and everything is nice and you can look out of the window and see the blue sky, or the answer is wrong and you have to start over and try again and see how it comes out this time.

Carl Sandburg (1878-1967)

3.1 Introduction

IN SYSTEM IDENTIFICATION applications such as acoustic echo cancellation (AEC) shown in Fig. 2.1, an FIR adaptive filter is used to identify an unknown time-varying system that is assumed to be linear. Important performance measures for adaptive filters characterize the initial convergence rate, the residual error after convergence, the ability to track time-varying systems and the computational complexity. This chapter focuses on analyzing the steady-state misalignment performances of a class of MMax-based algorithms including MMax-NLMS, MMax-AP and MMax-RLS as discussed in Chapter 2, when tracking time-varying systems. Consideration of an algorithm performance under such

dynamic conditions is important since, in the applications of interest, the unknown system is often continuously time-varying. It is therefore necessary to include a time-varying system model in the analysis of such adaptive algorithms as indicated in several significant studies including [73] [74] [75] [76].

The time-varying channel model in [77] which uses a modified first-order Markov model of the unknown system is adopted for analysis. Whereas the work in [77] specifically addresses LMS and RLS, analysis framework presented in this chapter extends that work to a more general form that can be applied to a wider range of adaptive algorithms including NLMS, AP, RLS and, in particular, the MMax selective-tap algorithms that is the main focus. Through this analysis, this chapter presents new insights into the tracking capability of selective-tap algorithms by highlighting and comparing the performances for a class of fully updating algorithms and their MMax variants under both time-invariant and time-varying unknown system conditions. It is shown, for each algorithm, how the tracking performance is degraded by the MMax tap selection and the degradation in steady-state misalignment performance is quantified analytically under common assumptions.

This chapter is organized as follows: The modified first-order Markov model [77] used for the time-varying unknown system is reviewed in Section 3.2 while Section 3.3 develops a general analysis framework for steady-state misalignment in a time-varying unknown system condition case. Having established the new analysis framework and applied it to standard adaptive filtering examples, the principal contribution of Section 3.4 is the steady-state misalignment analysis of selective-tap MMax algorithms. The analysis of MMax-NLMS includes Max-NLMS [65] as a special case. Comparative results are shown in Section 3.5 to verify the analytically derived misalignment performance against simulation learning curves for single channel AEC. In this chapter, for reason of compactness, the dependency of a variable on sample iteration n is denoted as a subscript giving \mathbf{h}_n for the unknown impulse response, such that at each sample iteration, the l^{th} element of this vector is now denoted as $h_n(l)$, i.e.,

$$\mathbf{h}_n = [h_n(0) \ h_n(1) \ \dots \ h_n(L-1)]^T . \quad (3.1)$$

3.2 Non-stationary system model

The modified first-order Markov model [77] is employed to represent a time-varying unknown system

$$\mathbf{h}_n = \xi \mathbf{h}_{n-1} + \sqrt{1 - \xi^2} \mathbf{s}_n, \quad (3.2)$$

where

$$\mathbf{h}_n = [h_n(0) \ h_n(1) \ \dots \ h_n(L_R - 1)]^T \quad (3.3)$$

is the impulse response of the unknown system with length L_R and

$$\mathbf{s}_n = [s_n(0) \ s_n(1) \ \dots \ s_n(L_R - 1)]^T \quad (3.4)$$

is an uncorrelated noise process with elements drawn from a normal (Gaussian) distribution with zero mean and variance σ_s^2 . This model has the key features that (i) the single parameter $0 \ll \xi < 1$ controls the relative contributions to the instantaneous values of the coefficients of “system memory” (the term $\xi \mathbf{h}_{n-1}$) and “innovations” (the term $\sqrt{1 - \xi^2} \mathbf{s}_n$), (ii) the average power of the norm of the coefficients is independent of ξ .

Defining system change as

$$\begin{aligned} \Delta \mathbf{h}_n &= \mathbf{h}_n - \mathbf{h}_{n-1} \\ &= -(1 - \xi) \mathbf{h}_{n-1} + \sqrt{1 - \xi^2} \mathbf{s}_n, \end{aligned} \quad (3.5)$$

and assuming [77] [78]

$$E\{\|\mathbf{h}_{n-1}\|_2^2\} = L\sigma_s^2, \quad (3.6)$$

where $\|\cdot\|_2^2$ and $E\{\cdot\}$ are defined as the l_2 -norm and mathematical expectation operator

respectively, it follows that

$$\begin{aligned}
 E\{\|\mathbf{h}_n\|_2^2\} &= \xi^2 E\{\|\mathbf{h}_{n-1}\|_2^2\} + (1 - \xi^2)L\sigma_s^2 \\
 &= \xi^2 L\sigma_s^2 + (1 - \xi^2)L\sigma_s^2 \\
 &= L\sigma_s^2 .
 \end{aligned} \tag{3.7}$$

In the limit $n \rightarrow \infty$, the mean square change of the unknown system is then given by

$$\begin{aligned}
 \lim_{n \rightarrow \infty} E\{\|\Delta \mathbf{h}_n\|_2^2\} &= (1 - \xi)^2 E\{\|\mathbf{h}_{n-1}\|_2^2\} + (1 - \xi^2)L\sigma_s^2 \\
 &= 2L\sigma_s^2(1 - \xi) ,
 \end{aligned} \tag{3.8}$$

which is a monotonically decreasing function of ξ and is proportional to L and variance of \mathbf{s}_n . As will be shown through an experimental illustration in Section 3.5, for $\xi = 0.9999$ and $\sigma_s^2 = 1$ given in (3.2), the tracking performance of the NLMS algorithm is comparatively equivalent to the algorithm tracking a source moving at 0.2 ms^{-1} for acoustic impulse responses \mathbf{h}_n generated using the method of images [72] with an adaptive filter length of $L = L_R = 64$.

3.3 General misalignment analysis for time-varying systems

Adaptive algorithms of the form

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} e_{n-k} \tag{3.9}$$

are considered, where K is defined as the projection order,

$$\mathbf{x}_{n-k} = [x_{n-k}(0) \ x_{n-k}(1) \ \dots \ x_{n-k}(L-1)]^T \tag{3.10}$$

is the tap-input vector at iteration $n - k$. In this chapter, for mathematical tractability, elements in \mathbf{x}_{n-k} are drawn from a zero mean white noise process with Gaussian distribution

Table 3.1: Projection order and Γ_{n-k} for various algorithms

Algorithm	Projection order	Γ_{n-k}	c
LMS	$K = 1$	$c\mathbf{I}_{L \times L}$	2μ
NLMS	$K = 1$	$c\mathbf{I}_{L \times L}$	$\frac{2\mu}{\mathbf{x}_n^T \mathbf{x}_n}$
AP	K	$c\mathbf{I}_{L \times L}$	$\frac{2\mu}{\mathbf{x}_{n-k}^T \mathbf{x}_{n-k}}$
RLS	$K = 1$	c	Ψ_n^{-1}

and variance σ_x^2 . The estimated impulse response $\hat{\mathbf{h}}_n$ is given by

$$\hat{\mathbf{h}}_n = [\hat{h}_n(0) \hat{h}_n(1) \dots \hat{h}_n(L-1)]^T \quad (3.11)$$

while e_{n-k} is the *a priori* error given by

$$e_{n-k} = y_{n-k} - \hat{\mathbf{h}}_{n-k-1}^T \mathbf{x}_{n-k} \quad (3.12)$$

with y_{n-k} being the received microphone signal as depicted in Fig. 2.1. Using the generalized update form given in (3.9) and defining $\mathbf{I}_{L \times L}$ as a $L \times L$ identity matrix, the $L \times L$ adaptation control matrix Γ_{n-k} is defined in Table 3.1 for the respective algorithms. Note that for the AP algorithm with

$$\mathbf{X}_{a,n} = [\mathbf{x}_n \ \mathbf{x}_{n-1} \ \dots \ \mathbf{x}_{n-K+1}]^T \quad (3.13)$$

as defined in (2.52) and in a similar approach to [79] [80], for $1 < K \ll L$, it is assumed

$$\mathbf{X}_{a,n} \mathbf{X}_{a,n}^T = \text{diag}\{\|\mathbf{x}_n\|_2^2 \ \|\mathbf{x}_{n-1}\|_2^2 \ \dots \ \|\mathbf{x}_{n-K+1}\|_2^2\} \quad (3.14)$$

hence giving

$$[\mathbf{X}_{a,n}\mathbf{X}_{a,n}^T]^{-1} = \begin{bmatrix} \frac{1}{\|\mathbf{x}_n\|_2^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{\|\mathbf{x}_{n-1}\|_2^2} & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \frac{1}{\|\mathbf{x}_{n-K+1}\|_2^2} \end{bmatrix}. \quad (3.15)$$

For the purpose of this analysis, it is assumed that $E\{\mathbf{h}_n\} = \mathbf{0}_{L \times 1}$ where $\mathbf{0}_{L \times 1}$ is a $L \times 1$ null vector. In addition, as explained in Section 2.2.2, to neglect any additional misalignment effects due to undermodelling, the dimension of the estimated impulse response $\hat{\mathbf{h}}_n$ has been chosen to match the dimension of the unknown impulse response \mathbf{h}_n , i.e., $L = L_R$. Defining the system mismatch vector

$$\mathbf{v}_n = \hat{\mathbf{h}}_n - \mathbf{h}_n, \quad (3.16)$$

the *a priori* error is then given by

$$e_n = w_n - \mathbf{x}_n^T \mathbf{v}_{n-1}, \quad (3.17)$$

where measurement noise w_n is an uncorrelated white noise sequence with Gaussian distribution (WGN) as depicted in Fig. 2.1 with $E\{w_n\} = 0$. Using (3.2), (3.9) and (3.17),

the system mismatch vector \mathbf{v}_n can be expressed as

$$\begin{aligned}
\mathbf{v}_n &= \hat{\mathbf{h}}_n - \mathbf{h}_n \\
&= \hat{\mathbf{h}}_{n-1} + \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} e_{n-k} - \mathbf{h}_n \\
&= \hat{\mathbf{h}}_{n-1} - \mathbf{h}_n + \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} w_{n-k} - \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} \\
&= \hat{\mathbf{h}}_{n-1} - \xi \mathbf{h}_{n-1} + \mathbf{h}_{n-1} - \mathbf{h}_n + \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} w_{n-k} \\
&\quad - \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} - \sqrt{1 - \xi^2} \mathbf{s}_n \\
&= \mathbf{v}_{n-1} + (1 - \xi) \mathbf{h}_{n-1} + \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} w_{n-k} \\
&\quad - \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} - \sqrt{1 - \xi^2} \mathbf{s}_n .
\end{aligned} \tag{3.18}$$

Using the independence theory¹ [35] [81], the system mismatch autocorrelation matrix $\mathbf{R}_{\mathbf{v},n}$ can be expressed as

$$\begin{aligned}
\mathbf{R}_{\mathbf{v},n} &= E\{\mathbf{v}_n \mathbf{v}_n^T\} \\
&= \mathbf{R}_{\mathbf{v},n-1} + 2(1 - \xi) \sigma_s^2 \mathbf{I} + K \sigma_w^2 \sum_{k=0}^{K-1} E\{\Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \Gamma_{n-k}^T\} \\
&\quad - \mathbf{R}_{\mathbf{v},n-1} \sum_{k=0}^{K-1} E\{\Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T\} - \mathbf{R}_{\mathbf{v},n-1} \sum_{k=0}^{K-1} E\{\mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \Gamma_{n-k}^T\} \\
&\quad + E\left\{ \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} \sum_{r=0}^{K-1} \mathbf{v}_{n-r-1}^T \mathbf{x}_{n-r} \mathbf{x}_{n-r}^T \Gamma_{n-r}^T \right\} ,
\end{aligned} \tag{3.19}$$

¹The independence theory imposes certain conditions on the data for mathematical tractability. It is assumed that (i) the input sequence \mathbf{x}_n is drawn from an independent and identically distributed (i.i.d.) process, (ii) y_n is independent on \mathbf{x}_m for $n > m$ and (iii) noise sequence w_n is also i.i.d. and statistically independent of \mathbf{x}_n .

where

$$\begin{aligned}
E\{\mathbf{v}_{n-1}\mathbf{v}_{n-1}^T\} &= \mathbf{R}_{\mathbf{v},n-1}, \\
E\{w_{n-k}w_{n-r}\} &= \begin{cases} \sigma_w^2, & k=r, \\ 0, & \text{otherwise,} \end{cases} \\
E\{w_{n-k}\mathbf{v}_{n-1}^T\} &= \mathbf{0}_{1\times L}, \\
E\{w_{n-k}\mathbf{h}_{n-1}^T\} &= \mathbf{0}_{1\times L}, \\
E\{\mathbf{s}_n\mathbf{v}_{n-1}^T\} &= \mathbf{0}_{L\times L}, \\
E\{w_{n-k}\mathbf{s}_n^T\} &= \mathbf{0}_{1\times L}, \\
E\{\mathbf{s}_n\mathbf{h}_{n-1}^T\} &= \mathbf{0}_{L\times L}, \\
E\{\mathbf{h}_{n-1}\mathbf{v}_{n-1}^T\} &= \mathbf{0}_{L\times L}, \\
E\left\{\mathbf{v}_{n-1}\sum_{k=0}^{K-1}\mathbf{v}_{n-k-1}^T\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\Gamma_{n-k}^T\right\} &= \mathbf{R}_{\mathbf{v},n-1}\sum_{k=0}^{K-1}E\left\{\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\Gamma_{n-k}^T\right\},
\end{aligned}$$

and from (3.7),

$$\begin{aligned}
E\{\mathbf{h}_n\mathbf{h}_n^T\} &= E\{\mathbf{s}_n\mathbf{s}_n^T\} \\
&= \sigma_s^2\mathbf{I}_{L\times L}
\end{aligned} \tag{3.20}$$

have been employed. Following the approach adopted in [35], it has also been assumed that for large n , the time variations of the system mismatch vector \mathbf{v}_n , are sufficiently slow compared to those of the input vector \mathbf{x}_n since the adaptive filter is able to track the unknown system to within a time lag and as a consequence, \mathbf{v}_n is independent to \mathbf{x}_n and after convergence, $\mathbf{v}_n \approx \mathbf{v}_{n-k}$ while

$$E\{\mathbf{v}_{n-k}\mathbf{v}_{n-k}^T\} \approx E\{\mathbf{v}_n\mathbf{v}_n^T\} = \mathbf{R}_{\mathbf{v},n}. \tag{3.21}$$

Under these assumptions, the autocorrelation matrix of the system mismatch vector which is approximately time-invariant is then denoted as $\mathbf{R}_{\mathbf{v}}$. Employing the normalized mis-

alignment as defined in (2.11) and reproduced here for convenience

$$\begin{aligned}\eta &= \frac{\|\mathbf{h}_n - \widehat{\mathbf{h}}_n\|_2^2}{\|\mathbf{h}_n\|_2^2} \\ &= \frac{\eta'}{\|\mathbf{h}_n\|_2^2},\end{aligned}\tag{3.22}$$

the steady-state misalignment can be expressed, for large n , as $\eta' = \text{tr}\{\mathbf{R}_v\}$ where $\text{tr}\{\cdot\}$ is the trace operator.

3.3.1 Mean square misalignment with $K = 1$ and $M = L$ for NLMS and RLS

A fully updated algorithm is initially considered in this section where $\Gamma_n = \Gamma, \forall n$, is time-invariant, and tap-input vector \mathbf{x}_n is drawn from a zero mean white noise process with Gaussian distribution with variance σ_x^2 . Using the factorization property of independent Gaussian variables [82] as shown in Section 3.7.1 and denoting

$$\mathbf{R}_{xx} = E\{\mathbf{x}_n \mathbf{x}_n^T\}\tag{3.23}$$

as the autocorrelation matrix of the input signal, the expectations in (3.19) can be evaluated for projection order $K = 1$ using

$$E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} = \Gamma \mathbf{R}_{xx} \Gamma,\tag{3.24a}$$

$$E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T\} = \Gamma \mathbf{R}_{xx},\tag{3.24b}$$

$$E\{\mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} = \mathbf{R}_{xx} \Gamma,\tag{3.24c}$$

$$E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} = \Gamma \left[2\mathbf{R}_{xx} \mathbf{R}_{v,n-1} \mathbf{R}_{xx} + \mathbf{R}_{xx} \text{tr}\{\mathbf{R}_{xx} \mathbf{R}_{v,n-1}\} \right] \Gamma.\tag{3.24d}$$

Substituting (3.20) and (3.24a)-(3.24d) into (3.19) gives

$$\begin{aligned} \mathbf{R}_{\mathbf{v},n} &= \mathbf{R}_{\mathbf{v},n-1} - \mathbf{R}_{\mathbf{v},n-1}\Gamma\mathbf{R}_{\mathbf{xx}} - \mathbf{R}_{\mathbf{v},n-1}\mathbf{R}_{\mathbf{xx}}\Gamma \\ &\quad + \Gamma\left[2\mathbf{R}_{\mathbf{xx}}\mathbf{R}_{\mathbf{v},n-1}\mathbf{R}_{\mathbf{xx}} + \mathbf{R}_{\mathbf{xx}}\text{tr}\{\mathbf{R}_{\mathbf{xx}}\mathbf{R}_{\mathbf{v},n-1}\}\right]\Gamma \\ &\quad + \Gamma\mathbf{R}_{\mathbf{xx}}\Gamma\sigma_w^2 + 2(1-\xi)\sigma_s^2\mathbf{I}_{L\times L}. \end{aligned} \quad (3.25)$$

The steady-state misalignment η' can be found by first considering $\Gamma = c\mathbf{I}_{L\times L}$ and white Gaussian noise (WGN) input with variance σ_x^2 giving $\mathbf{R}_{\mathbf{xx}} = \sigma_x^2\mathbf{I}_{L\times L}$. The variable c in Γ is a scalar quantity specific for each algorithm which will be described in the sequel. Assuming the system mismatch error \mathbf{v}_n is fluctuating around its mean, the system mismatch autocorrelation matrix $\mathbf{R}_{\mathbf{v},n}$ in (3.25) can be simplified and the steady-state misalignment $\eta' = \text{tr}\{\mathbf{R}_{\mathbf{v}}\}$ is given by

$$\begin{aligned} \eta' &= \eta' - 2c\sigma_x^2\eta' + 2\sigma_x^4c^2\eta' + \sigma_x^4c^2L\eta' \\ &\quad + c^2\sigma_x^2\sigma_w^2L + 2(1-\xi)\sigma_s^2L \end{aligned}$$

from which can then be expressed as

$$\eta' = \frac{c\sigma_w^2L}{2\phi} + \frac{(1-\xi)L\sigma_s^2}{c\sigma_x^2\phi}, \quad (3.26)$$

where

$$\phi = 1 - c\sigma_x^2\left(1 + \frac{L}{2}\right) \quad (3.27)$$

and c is an algorithm dependent term. Adopting the terminology of [76], the first term in (3.26) corresponds to the *estimation variance* and is dependent on measurement noise w_n and the second term in (3.26) corresponds to the *lag variance* and is due to system time variation ξ . It can also be seen from (3.26) that the estimation and lag variances are uncoupled.

For the LMS case, $c = 2\mu$ giving

$$\eta'_{\text{LMS}} = \frac{\mu\sigma_w^2L}{\phi} + \frac{(1-\xi)L\sigma_s^2}{2\mu\sigma_x^2\phi}. \quad (3.28)$$

The estimation variance term of this result is, as expected, proportional to μ and consistent with that presented in [35] for which it is assumed $\phi \approx 1$ for small μ . However, the analysis presented here needs no such assumption. The lag variance term is inversely proportional to μ and linearly dependent on the system variation parameter ξ .

For NLMS, $c = 2\mu/(L\sigma_x^2)$ giving

$$\eta'_{\text{NLMS}} = \frac{\mu\sigma_w^2}{\sigma_x^2\phi} + \frac{(1-\xi)L^2\sigma_s^2}{2\mu\phi}. \quad (3.29)$$

It is interesting, from a step-size control point of view, to evaluate the step-size which achieves the lowest misalignment μ_{mis} by differentiating η'_{NLMS} in (3.29) with respect to step-size μ to obtain

$$\frac{d \eta'_{\text{NLMS}}}{d \mu} = \frac{\sigma_w^2}{\sigma_x^2} \left[\frac{\varsigma\mu}{(1-\varsigma\mu)^2} + \frac{1}{1-\varsigma\mu} \right] + \frac{(1-\xi)L^2\sigma_s^2}{2} \left[\frac{2\varsigma\mu - 1}{\mu^2(1-\varsigma\mu)^2} \right], \quad (3.30)$$

where

$$\varsigma = 2(1 + L/2)/L. \quad (3.31)$$

Setting $d \eta'_{\text{NLMS}}/d \mu = 0$, a quadratic equation in terms of μ_{mis} given by

$$\frac{\sigma_w^2}{\sigma_x^2} \mu_{\text{mis}}^2 + (1-\xi)L^2\sigma_s^2\varsigma\mu_{\text{mis}} - \frac{(1-\xi)L^2\sigma_s^2}{2} = 0 \quad (3.32)$$

is obtained. Under the condition that $0 < \mu_{\text{mis}} \leq 1$, the step-size giving the lowest misalignment under non-stationary unknown system condition for NLMS is given by

$$\mu_{\text{mis}} = 0.5 \frac{\sigma_x^2}{\sigma_w^2} \left[- (1-\xi)L^2\sigma_s^2\varsigma + \sqrt{\left[(1-\xi)L^2\sigma_s^2\varsigma \right]^2 + 2 \left(\frac{\sigma_w^2}{\sigma_x^2} \right) (1-\xi)L^2\sigma_s^2} \right]. \quad (3.33)$$

Figure 3.1 illustrates the variation of μ_{mis} with ξ under various signal-to-noise ratio (SNR) conditions. The SNR is computed using w_n and y_n where the latter is obtained by $\mathbf{h}_n^T \mathbf{x}_n$ as shown in Fig. 2.1. The parameters for this illustrative example are $L = 128$, $\sigma_s^2 = 1$ and $\sigma_x^2 = 0.962$. For each case of SNR, the well-known result that for reducing system variation $\xi \rightarrow 1$, $\mu_{\text{mis}} \rightarrow 0$ can be observed and hence a smaller step-size achieves a lower steady-state misalignment, though at the expense of reduced convergence

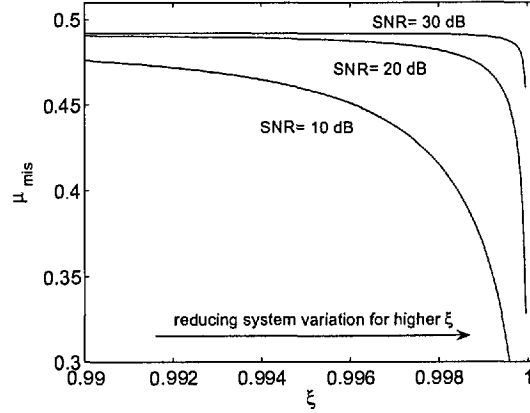


Figure 3.1: Variation of μ_{mis} with ξ under various SNR conditions.

rate. Under time-varying unknown system conditions $\xi < 1$, it can be seen as expected that μ_{mis} increases smoothly as ξ reduces, since for higher time-varying unknown system condition, step-size μ must be sufficiently high for tracking. In addition, for any given ξ , μ_{mis} increases with SNR. As will be seen through simulation examples in Section 3.5.5, under the condition $\xi < 1$, performance of NLMS in terms of convergence rate and steady-state misalignment increases with μ within the region $0 < \mu \leq \mu_{\text{mis}}$.

For RLS, c in

$$\Gamma = c\mathbf{I}_{L \times L} = \Psi_n^{-1} \quad (3.34)$$

can be determined by considering the time-averaged autocorrelation matrix Ψ_n defined in (2.56) and is reproduced here for convenience

$$\Psi_n = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}_i \mathbf{x}_i^T. \quad (3.35)$$

In the limit $n \rightarrow \infty$

$$\begin{aligned} E\left\{ \lim_{n \rightarrow \infty} \Psi_n \right\} &= E\left\{ \lim_{n \rightarrow \infty} (\lambda^{n-1} \mathbf{x}_1 \mathbf{x}_1^T + \lambda^{n-2} \mathbf{x}_2 \mathbf{x}_2^T + \dots + \mathbf{x}_n \mathbf{x}_n^T) \right\} \\ &= \lim_{n \rightarrow \infty} (\lambda^{n-1} + \lambda^{n-2} + \dots + 1) \mathbf{R}_{\mathbf{x}\mathbf{x}} \\ &= \frac{1}{1-\lambda} \mathbf{R}_{\mathbf{x}\mathbf{x}}, \end{aligned} \quad (3.36)$$

where \mathbf{R}_{xx} is the true input signal autocorrelation matrix defined in (3.23). Using a quasi-deterministic approximation for large n [35], $\Psi_n \approx \mathbf{R}_{xx}/(1-\lambda)$ and hence

$$\Gamma = \Psi_n^{-1} \approx (1-\lambda)\mathbf{R}_{xx}^{-1}. \quad (3.37)$$

For the case when the input signal, x_n , is drawn from a white noise process with Gaussian distribution, then from (3.34), the scalar constant $c = (1-\lambda)/\sigma_x^2$. Using (3.26), the steady-state misalignment is given as

$$\eta'_{\text{RLS}} \approx \frac{(1-\lambda)L\sigma_w^2}{2\sigma_x^2\phi} + \frac{(1-\xi)L\sigma_s^2}{(1-\lambda)\phi}, \quad (3.38)$$

where the term ϕ is defined in (3.27). By taking the derivative of steady-state misalignment η'_{RLS} with respect to the forgetting factor λ , the well-known result for a time-invariant unknown system condition $\xi = 1$,

$$\frac{d \eta'_{\text{RLS}}}{d \lambda} = -\frac{L\sigma_w^2}{2\sigma_x^2\phi^2} < 0 \quad (3.39)$$

can be seen. Hence, the steady-state misalignment is a decreasing function of the forgetting factor λ , although for a smaller λ the rate of convergence is increased.

Furthermore, the effect of ξ on steady-state misalignment η'_{RLS} can be analyzed by first differentiating η'_{RLS} with respect to ξ and finding the boundary condition for λ . Assuming

$$\frac{d \eta'_{\text{RLS}}}{d \xi} = \frac{L\sigma_s^2}{(1-\lambda)\phi} < 0, \quad (3.40)$$

and noting that $L\sigma_s^2 > 0$ and $1-\lambda > 0$, the condition

$$\lambda > 1 - [1/(1+L/2)] \quad (3.41)$$

can be obtained. For a typical range [15], $1 - 1/(3L) \leq \lambda \leq 1 - 1/(10L)$ and since $L > 1$, it can be seen that the conditions (3.41) and consequently (3.40) are satisfied. Hence as will be shown through simulation examples in Section 3.5.4, the steady-state misalignment η'_{RLS} reduces for a lower system variation as $\xi \rightarrow 1$.

3.3.2 Mean square misalignment with $K \neq 1$ and $M = L$ for AP

This analysis can be applied to the AP algorithm for the condition $1 \leq K \ll L$. With reference to Table 3.1 and for a WGN input sequence \mathbf{x}_n with variance σ_x^2 , $\mathbf{x}_{n-k}^T \mathbf{x}_{n-k} = L\sigma_x^2$, $\forall n, k$, hence giving $\Gamma_{n-k} = \Gamma$. Exploiting the linear property of the expectation operator, the terms in (3.19) can be evaluated using the following relations

$$\sum_{k=0}^{K-1} E \left\{ \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \Gamma_{n-k}^T \right\} = K \Gamma \mathbf{R}_{\mathbf{xx}} \Gamma, \quad (3.42a)$$

$$\sum_{k=0}^{K-1} E \left\{ \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \right\} = K \Gamma \mathbf{R}_{\mathbf{xx}}, \quad (3.42b)$$

$$\sum_{k=0}^{K-1} E \left\{ \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \Gamma_{n-k}^T \right\} = K \mathbf{R}_{\mathbf{xx}} \Gamma. \quad (3.42c)$$

Following a similar approach to (3.24d) where the factorization property of independent Gaussian variables [82] as shown in Section 3.7.1 is employed, the last term of (3.19) can be simplified as follows

$$\begin{aligned} & E \left\{ \sum_{k=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} \sum_{r=0}^{K-1} \mathbf{v}_{n-r-1}^T \mathbf{x}_{n-r} \mathbf{x}_{n-r}^T \Gamma_{n-r}^T \right\} \\ &= E \left\{ \sum_{k=0}^{K-1} \sum_{r=0}^{K-1} \Gamma_{n-k} \mathbf{x}_{n-k} \mathbf{x}_{n-k}^T \mathbf{v}_{n-k-1} \mathbf{v}_{n-r-1}^T \mathbf{x}_{n-r} \mathbf{x}_{n-r}^T \Gamma_{n-r}^T \right\} \\ &= K^2 \Gamma \left[2 \mathbf{R}_{\mathbf{xx}} \mathbf{R}_{\mathbf{v},n-1} \mathbf{R}_{\mathbf{xx}} + \mathbf{R}_{\mathbf{xx}} \text{tr} \{ \mathbf{R}_{\mathbf{xx}} \mathbf{R}_{\mathbf{v},n-1} \} \right] \Gamma. \end{aligned} \quad (3.43)$$

Substituting (3.42a)-(3.42c) and (3.43) into (3.19), the system mismatch autocorrelation matrix $\mathbf{R}_{\mathbf{v},n}$ can be expressed as

$$\begin{aligned} \mathbf{R}_{\mathbf{v},n} &= \mathbf{R}_{\mathbf{v},n-1} + 2(1-\xi)\sigma_s^2 \mathbf{I}_{L \times L} + K^2 \sigma_w^2 \Gamma \mathbf{R}_{\mathbf{xx}} \Gamma^T \\ &\quad - K \mathbf{R}_{\mathbf{v},n-1} \Gamma \mathbf{R}_{\mathbf{xx}} - K \mathbf{R}_{\mathbf{v},n-1} \mathbf{R}_{\mathbf{xx}} \Gamma \\ &\quad + K^2 \Gamma \left[2 \mathbf{R}_{\mathbf{xx}} \mathbf{R}_{\mathbf{v},n-1} \mathbf{R}_{\mathbf{xx}} + \mathbf{R}_{\mathbf{xx}} \text{tr} \{ \mathbf{R}_{\mathbf{xx}} \mathbf{R}_{\mathbf{v},n-1} \} \right] \Gamma. \end{aligned} \quad (3.44)$$

Similar to (3.25), it is assumed that $\mathbf{R}_{\mathbf{v},n}$ is fluctuating around its mean, $\Gamma = c \mathbf{I}_{L \times L}$ and $\mathbf{R}_{\mathbf{xx}} = \sigma_x^2 \mathbf{I}_{L \times L}$. The steady-state misalignment η'_{AP} for the AP algorithm can be obtained

by simplifying (3.44) and using $c = 2\mu/(L\sigma_x^2)$ from which

$$\eta'_{\text{AP}} = \frac{K\mu\sigma_w^2}{\sigma_x^2\phi_a} + \frac{(1-\xi)\sigma_s^2L^2}{2\mu K\phi_a}, \quad (3.45)$$

where

$$\phi_a = 1 - Kc\sigma_x^2\left(1 + \frac{L}{2}\right). \quad (3.46)$$

Note that (3.46) is similar to (3.27) except for a projection order term K . Furthermore, when $K = 1$, ϕ reduces to ϕ_a and hence the steady-state misalignment of AP is equivalent to that of the NLMS algorithm as expected.

3.4 Misalignment analysis of algorithms employing MMax tap selection

3.4.1 Misalignment analysis of MMax-NLMS

Partial update NLMS algorithms have been analyzed in, for example [19] [21] [65] [83] [84]. In [21] [83], it has been shown that for $M = 1$ tap being selected for adaptation at each sample iteration, MMax-NLMS converges for a zero mean WGN input sequence under the condition $0 < \mu < L/(L + 2)$. In addition, the excess mean square error of MMax-NLMS is derived for the case of $M = 1$. In the following, the steady-state misalignment is derived for an arbitrary case of $M < L$.

The MMax-NLMS [66] algorithm is characterized by (3.9) for projection order $K = 1$ with the $L \times L$ adaptation control matrix given by

$$\Gamma_n = \mu_n \mathbf{Q}_n \quad (3.47)$$

in which the elements of the MMax diagonal matrix $\mathbf{Q}_n = \text{diag}\{\mathbf{q}_n\}$ are determined from (2.47) and $\tilde{\mathbf{x}}_n = \mathbf{Q}_n \mathbf{x}_n$ is the subselected tap-input vector. The variable μ_n is a scalar constant specific to each algorithm as will be discussed in the sequel.

For convergence in the mean square, consider (3.19) and the evaluation of $E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T\}$

and that tap selection elements $q_n(i)$, for $i = 0, 1, \dots, L-1$, are not independent of $x_n(i)$ as they ensure that only the M largest $|x_n(i)|$ are selected. The M selected samples are assumed to have zero mean and exploiting the mean ergodic theorem² [35], the variance of $\tilde{\mathbf{x}}_n$ is

$$\tilde{\sigma}_x^2 = \frac{1}{L} \sum_{i=0}^{L-1} \tilde{x}^2(n-i) \quad (3.48)$$

where

$$\begin{aligned} \tilde{\mathbf{x}}_n &= \mathbf{Q}_n \mathbf{x}_n \\ &= [\tilde{x}(n) \tilde{x}(n-1) \dots \tilde{x}(n-L+1)]^T \end{aligned} \quad (3.49)$$

is the subselected tap-input vector and that for $i = 0, 1, \dots, L-1$, some elements $\tilde{x}(n-i) = 0$ due to tap selection. Assuming that $\mathbf{x}_n \mathbf{x}_n^T$ is diagonal and using $E\{\mu_n\} = c$, a scalar constant such that

$$\begin{aligned} E\{\Gamma_n\} &= E\{\mu_n\} E\{\mathbf{Q}_n\} \\ &= c E\{\mathbf{Q}_n\}, \end{aligned} \quad (3.50)$$

the terms $E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T\}$ and $E\{\mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\}$ can be simplified as

$$\begin{aligned} E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T\} &= E\{\mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} \\ &= E\{\mu_n\} E\{\mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T\} \\ &= \frac{M}{L} c \tilde{\sigma}_x^2 \mathbf{I}_{L \times L}. \end{aligned} \quad (3.51)$$

The condition $E\{\Gamma_n\} = \Gamma$ implicit in (3.24d) is not valid in this case. However, the

²The mean ergodic theorem states that if elements in \mathbf{x}_n is drawn from a stationary process which is valid for WGN inputs, then its time averages tends to $E\{\mathbf{x}_n\}$ as the length of the available sample n tends to ∞ . This theory can be extended to higher moments such as variance estimation as shown in [35] [85] [86].

evaluation of $\text{tr}\{\mathbf{R}_{\mathbf{v},n}\}$ in (3.19) can be achieved by using

$$\begin{aligned} \text{tr}\left\{E\left\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\right\}\right\} &= \text{tr}\left\{c^2 E\left\{\mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{x}_n \mathbf{x}_n^T\right\}\right\} \\ &= c^2 \text{tr}\left\{\mathbf{R}_{\mathbf{v},n-1}(L+2) \frac{M}{L} \tilde{\sigma}_x^2 \sigma_x^2 \mathbf{I}_{L \times L}\right\} \\ &= c^2 \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\}(L+2) \frac{M}{L} \tilde{\sigma}_x^2 \sigma_x^2, \end{aligned} \quad (3.52)$$

$$\begin{aligned} \text{tr}\left\{E\left\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\right\}\right\} &= \text{tr}\left\{\frac{M}{L} c^2 \tilde{\sigma}_x^2 \mathbf{I}_{L \times L}\right\} \\ &= \frac{M}{L} c^2 \tilde{\sigma}_x^2 L. \end{aligned} \quad (3.53)$$

Substituting (3.51)-(3.53) and (3.20) into (3.19) and letting $K = 1$, the expression

$$\begin{aligned} \text{tr}\{\mathbf{R}_{\mathbf{v},n}\} &= \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\} - 2\text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\} \frac{M}{L} c \tilde{\sigma}_x^2 + c^2 \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\}(L+2) \frac{M}{L} \tilde{\sigma}_x^2 \sigma_x^2 \\ &\quad + \frac{M}{L} c^2 \tilde{\sigma}_x^2 \sigma_w^2 L + 2(1-\xi) \sigma_s^2 L \\ &= \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\} \left[1 - 2 \frac{M}{L} c \tilde{\sigma}_x^2 + (L+2) \frac{M}{L} c^2 \tilde{\sigma}_x^2 \sigma_x^2\right] \\ &\quad + M c^2 \tilde{\sigma}_x^2 \sigma_w^2 + 2(1-\xi) L \sigma_s^2 \end{aligned} \quad (3.54)$$

is obtained.

Contraction mapping concept has been introduced in [87] [88] [89] for adaptive algorithms. A contraction mapping is produced when the norm of the difference of the mapped vectors is less than the norm of the difference of the original vectors, i.e.,

$$\|\mathcal{T} \varrho_1 - \mathcal{T} \varrho_2\|_2 \leq C \|\varrho_1 - \varrho_2\|_2 \quad (3.55)$$

where $\|\cdot\|_2$ is defined as the l_2 -norm, $C < 1$ is a scalar, \mathcal{T} is a mapping operator and ϱ_1 and ϱ_2 are vector or scalar quantities. Applying the contraction mapping concept and following the same approach as [89], convergence for MMax-NLMS can be shown [71]

using (3.54) by letting

$$\begin{aligned}\mathcal{T} &= 1 - 2\frac{M}{L}c\tilde{\sigma}_x^2 + (L+2)\frac{M}{L}c^2\tilde{\sigma}_x^2\sigma_x^2, \\ \varrho_1 &= \text{tr}\{\mathbf{R}_{\mathbf{v},n}\}, \\ \varrho_2 &= \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\},\end{aligned}$$

from which substituting into (3.55) with the condition that $C < 1$ gives

$$\mathcal{T} = \left| 1 - 2\frac{M}{L}c\tilde{\sigma}_x^2 + (L+2)\frac{M}{L}c^2\tilde{\sigma}_x^2\sigma_x^2 \right| < 1 \quad (3.56)$$

and the convergence speed is faster for smaller values of \mathcal{T} . It can therefore be seen from (3.56) that, for typical values of $c \ll 1$, maximum convergence speed will be when all filter coefficients are updated at each sample iteration, i.e., $M = L$. Therefore MMax-NLMS suffers a decrease in convergence speed proportionate to M/L as compared to NLMS. As shown in Section 3.7.2, for the case of MMax-NLMS, $c = 2\mu/(L\sigma_x^2)$ giving

$$\mathcal{T} = \left| 1 - 4\mu\frac{M\tilde{\sigma}_x^2}{L^2\sigma_x^2} + 4\mu^2\frac{(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} \right| < 1 \quad (3.57)$$

so that

$$0 < \mu < \frac{L}{L+2}. \quad (3.58)$$

The misalignment for MMax-NLMS can be found [78] from (3.54) and using the approach of (3.26) as

$$\text{tr}\{\mathbf{R}_{\mathbf{v}}\} \left[2\frac{M}{L}c\tilde{\sigma}_x^2 - (L+2)\frac{M}{L}c^2\tilde{\sigma}_x^2\sigma_x^2 \right] = Mc^2\tilde{\sigma}_x^2\sigma_w^2 + 2(1-\xi)L\sigma_s^2$$

resulting in

$$\text{tr}\{\mathbf{R}_{\mathbf{v}}\} = \frac{c\sigma_w^2L}{2 - (L+2)c\sigma_x^2} + \frac{2(1-\xi)L^2\sigma_s^2/M}{2c\tilde{\sigma}_x^2 - (L+2)c^2\tilde{\sigma}_x^2\sigma_x^2}. \quad (3.59)$$

For MMax-NLMS where $c = 2\mu/(L\sigma_x^2)$, the steady-state misalignment is then

$$\eta'_{\text{MMax-NLMS}} = \frac{\mu\sigma_w^2}{\sigma_x^2\phi} + \frac{L\sigma_x^2}{\tilde{\sigma}_x^2M} \frac{(1-\xi)L^2\sigma_s^2}{2\mu\phi}, \quad (3.60)$$

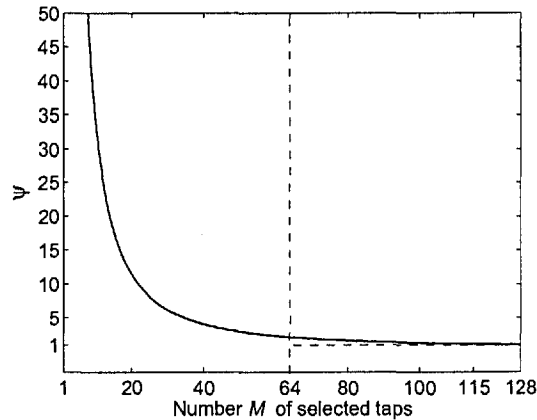


Figure 3.2: Variation of ψ with M selected coefficients per iteration for $\xi < 1$ showing a modest increment in ψ when $0.5L \leq M < L$.

where the term ϕ is defined in (3.27).

As can be seen, the *estimation variance* term is independent of M and is identical to that of NLMS. Thus for a time-invariant unknown system with $\xi = 1$, the steady-state normalized misalignment of the MMax-NLMS algorithm is the same as that of NLMS, i.e., $\eta'_{\text{MMax-NLMS}} = \eta'_{\text{NLMS}}$. Comparing (3.60) with (3.29), an additional factor arises in the lag variance of $L\sigma_x^2/(\tilde{\sigma}_x^2 M)$ for MMax-NLMS compared to NLMS. To quantify the closeness of tap selection to that of a full tap-input vector in an MMax sense, the M-ratio [27]

$$\mathcal{M} = \frac{\|\mathbf{Q}_n \mathbf{x}_n\|_2^2}{\|\mathbf{x}_n\|_2^2} = \frac{\tilde{\sigma}_x^2}{\sigma_x^2} \quad (3.61)$$

is defined. As shown in [27], \mathcal{M} exhibits only a modest reduction for $0.5L \leq M < L$ and hence a graceful reduction in convergence rate is expected over this range of M as compared to the fully updated NLMS algorithm. The variation of steady-state misalignment due to the M number of taps selected for updating can be analyzed by first noting from (3.60) that, under a time-varying unknown system condition $\xi < 1$, the lag variance is proportional to the term

$$\psi = L\sigma_x^2/(\tilde{\sigma}_x^2 M) = L/(M\mathcal{M}) . \quad (3.62)$$

Figure 3.2 shows the variation of ψ with the number of selected taps M for $L = 128$

using a zero mean, unit variance white Gaussian noise (WGN) input sequence. It can be observed that for full adaptation $M = L = 128$, $\psi = 1$ since $\tilde{\sigma}_x^2 = \sigma_x^2$ and $\mathcal{M} = 1$. More importantly, although ψ exhibits a factor of 2 increase when M is reduced from L to $0.5L$, this factor is insignificant compared to the case when M is reduced from $0.5L$ to 1. Hence for the range of $0.5L \leq M < L$, with reduced computational complexity, only an insignificant degradation in steady-state misalignment performance for time-varying case $\xi < 1$ is expected.

Similar to the NLMS algorithm discussed in Section 3.3.1, it is of interest to evaluate the step-size μ_{mis} which achieves the lowest misalignment by differentiating the steady-state misalignment $\eta'_{\text{MMax-NLMS}}$ in (3.60) with respect to step-size μ . Setting this differential equation to zero, the quadratic expression

$$\frac{\sigma_w^2}{\sigma_x^2} \mu_{\text{mis}}^2 + \psi(1 - \eta)L^2 \sigma_s^2 \varsigma \mu_{\text{mis}} - \frac{\psi(1 - \eta)L^2 \sigma_s^2}{2} = 0 \quad (3.63)$$

is obtained where

$$\varsigma = 2(1 + L/2)/L. \quad (3.64)$$

The step-size μ_{mis} for MMax-NLMS giving the lowest steady-state misalignment under non-stationary unknown system condition can then be obtained by noting $0 < \mu_{\text{mis}} \leq 1$ hence giving

$$\begin{aligned} \mu_{\text{mis}} = & 0.5 \frac{\sigma_x^2}{\sigma_w^2} \left[-\psi(1 - \xi)L^2 \sigma_s^2 \varsigma \right. \\ & \left. + \sqrt{\left[\psi(1 - \xi)L^2 \sigma_s^2 \varsigma \right]^2 + 2\psi \left(\frac{\sigma_w^2}{\sigma_x^2} \right) (1 - \xi)L^2 \sigma_s^2} \right]. \end{aligned} \quad (3.65)$$

As will be seen from simulations in Section 3.5.5, if $\mu_{\text{mis}} < \mu \leq 1$ under the non-stationary unknown system condition $\xi < 1$, convergence rate increases with step-size μ but at the expense of poorer steady-state misalignment. Consequently, for MMax-NLMS, the optimal step-size giving the highest rate of convergence while satisfying the minimum misalignment under non-stationary environment is μ_{mis} given in (3.65).

3.4.2 Misalignment analysis of MMax-RLS

Using (2.66) and (2.67), the update equation for the MMax-RLS algorithm may be expressed as

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \tilde{\Psi}_n^{-1} \mathbf{Q}_n \mathbf{x}_n e_n. \quad (3.66)$$

where the time-averaged subselected input autocorrelation matrix $\tilde{\Psi}_n$ is defined in (2.60) and \mathbf{Q}_n is the diagonal $L \times L$ MMax tap selection control matrix with elements defined in (2.47). In this case, projection order $K = 1$ in the general formulation of (3.9) and the tap selection control matrix Γ_n in (3.9) for MMax-RLS is given by

$$\Gamma_n = \tilde{\Psi}_n^{-1} \mathbf{Q}_n \quad (3.67)$$

where $\tilde{\Psi}_n$ can be expressed alternatively as

$$\tilde{\Psi}_n = \sum_{i=1}^n \lambda^{n-i} \mathbf{Q}_i \mathbf{x}_i \mathbf{x}_i^T \mathbf{Q}_i^T \quad (3.68)$$

with $0 \ll \lambda < 1$ being the forgetting factor. Following the same approach as for RLS in (3.36), with $n \rightarrow \infty$,

$$\begin{aligned} E \left\{ \lim_{n \rightarrow \infty} \tilde{\Psi}_n \right\} &= E \left\{ \lim_{n \rightarrow \infty} (\lambda^{n-1} \mathbf{Q}_1 \mathbf{x}_1 \mathbf{x}_1^T \mathbf{Q}_1^T + \dots + \mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{Q}_n^T) \right\} \\ &= \lim_{n \rightarrow \infty} \frac{M \tilde{\sigma}_x^2}{L} \left[\lambda^{n-1} \mathbf{I}_{L \times L} + \lambda^{n-2} \mathbf{I}_{L \times L} + \dots + \mathbf{I}_{L \times L} \right] \\ &= \frac{1}{1-\lambda} \frac{M \tilde{\sigma}_x^2}{L} \mathbf{I}_{L \times L}, \end{aligned} \quad (3.69)$$

and hence it follows from (3.67),

$$\Gamma_n = (1-\lambda) \frac{L}{M \tilde{\sigma}_x^2} \mathbf{Q}_n. \quad (3.70)$$

Employing (3.70), the following terms in (3.19) can be simplified as

$$\begin{aligned}
 E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T\} &= E\{\mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} \\
 &= (1-\lambda) \frac{L}{M\tilde{\sigma}_x^2} E\{\mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T\} \\
 &= (1-\lambda) \mathbf{I}_{L \times L}, \tag{3.71a}
 \end{aligned}$$

$$\begin{aligned}
 E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{v}_n \mathbf{v}_n^T \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} &= \left[\frac{(1-\lambda)L}{M\tilde{\sigma}_x^2} \right]^2 E\{\mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{Q}_n\} \mathbf{R}_{\mathbf{v},n} \\
 &= \frac{(1-\lambda)^2 L(L+2)\sigma_w^2}{M\tilde{\sigma}_x^2} \mathbf{R}_{\mathbf{v},n}, \tag{3.71b}
 \end{aligned}$$

$$\begin{aligned}
 E\{\Gamma_n \mathbf{x}_n \mathbf{x}_n^T \Gamma_n^T\} &= \left[\frac{(1-\lambda)L}{M\tilde{\sigma}_x^2} \right]^2 E\{\mathbf{Q}_n \mathbf{x}_n \mathbf{x}_n^T \mathbf{Q}_n\} \\
 &= \frac{(1-\lambda)^2 L}{M\tilde{\sigma}_x^2} \mathbf{I}_{L \times L}. \tag{3.71c}
 \end{aligned}$$

Following the same approach as (3.54), by substituting the set of equations (3.71a)-(3.71c) into (3.19) for projection order $K = 1$, the system mismatch autocorrelation matrix $\mathbf{R}_{\mathbf{v},n}$ can be expressed as

$$\begin{aligned}
 \mathbf{R}_{\mathbf{v},n} &= \mathbf{R}_{\mathbf{v},n-1} - 2(1-\lambda)\mathbf{R}_{\mathbf{v},n-1} + \frac{(1-\lambda)^2 L(L+2)\sigma_w^2}{M\tilde{\sigma}_x^2} \mathbf{R}_{\mathbf{v},n} \\
 &\quad + \frac{(1-\lambda)^2 L}{M\tilde{\sigma}_x^2} \sigma_w^2 \mathbf{I}_{L \times L} + 2(1-\xi)\sigma_s^2 \mathbf{I}_{L \times L}. \tag{3.72}
 \end{aligned}$$

As before, it is assumed that for large n , $\mathbf{R}_{\mathbf{v},n} \approx \mathbf{R}_{\mathbf{v}}$ is the approximately time-invariant autocorrelation matrix of the mean weight error vector. Defining steady-state misalignment vector $\eta' = \text{tr}\{\mathbf{R}_{\mathbf{v}}\}$ and using (3.72), the steady-state misalignment for the MMax-RLS algorithm can then be expressed as

$$\begin{aligned}
 \eta'_{\text{MMax-RLS}} &= \eta' - 2(1-\lambda)\eta' + \frac{(1-\lambda)^2 L(L+2)\sigma_w^2}{M\tilde{\sigma}_x^2} \eta' + \frac{(1-\lambda)^2 L^2 \sigma_w^2}{M\tilde{\sigma}_x^2} + 2L(1-\xi)\sigma_s^2 \\
 &= \frac{(1-\lambda)L^2 \sigma_w^2}{\beta_r} + \frac{2L(1-\xi)\sigma_s^2 M\tilde{\sigma}_x^2}{(1-\lambda)\beta_r}, \tag{3.73}
 \end{aligned}$$

where

$$\begin{aligned}
 \beta_r &= 2M\tilde{\sigma}_x^2 - (1-\lambda)L(L+2)\sigma_x^2 \\
 &= L\sigma_x^2 [2\psi^{-1} - (1-\lambda)(L+2)] \tag{3.74}
 \end{aligned}$$

and ψ is defined in (3.62).

Comparing (3.73) and (3.38), for full adaptation $M = L$, $\tilde{\sigma}_x^2 = \sigma_x^2$ and hence the steady-state misalignment for MMax-RLS is equivalent to that of RLS, i.e., $\eta'_{\text{MMax-RLS}} = \eta'_{\text{RLS}}$ as expected. More importantly, the estimation variance for MMax-RLS is dependent on the number of taps selected for adaptation M . As can be seen from (3.73) and (3.74), β_r is a decreasing function of ψ and hence for a time-invariant system with $\xi = 1$, the steady-state misalignment $\eta'_{\text{MMax-RLS}}$ is a decreasing function of M . This is contrary to $\eta'_{\text{MMax-NLMS}}$ in (3.60) where the steady-state misalignment is independent of M for time-invariant systems. Simulation results illustrating the dependency of the steady-state misalignment on M for MMax-RLS under the time-invariant unknown system condition $\xi = 1$ can be found in [24] and Section 3.5.4.

3.4.3 Misalignment analysis for MMax-AP

The update equation for the MMax-AP algorithm can be written in a similar form as (3.9) where for $k = 0, 1, \dots, K - 1$, the tap selection control matrix Γ_{n-k} can be expressed as

$$\Gamma_{n-k} = \mu_{n-k} \mathbf{Q}_{n-k}, \quad (3.75)$$

where $\mu_{n-k} = 2\mu / (\mathbf{x}_{n-k}^T \mathbf{x}_{n-k})$ is the step-size while elements of the MMax tap selection control matrix \mathbf{Q}_n are defined in (2.47). Assuming that the input signal is WGN, $E\{\mu_{n-k}\}$ can be evaluated, giving

$$E\{\mu_{n-k}\} = c = \frac{2\mu}{L\sigma_x^2}. \quad (3.76)$$

As before, the condition $E\{\Gamma_n\} = \Gamma$ is not valid here since elements in \mathbf{Q}_n is dependent on M and so $\text{tr}\{\mathbf{R}_{\mathbf{v},n}\}$ is evaluated using the same approach as (3.53) giving

$$\begin{aligned} \text{tr}\left\{K\sigma_w^2 \sum_{k=0}^{K-1} E\{\Gamma_{n-k}\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\Gamma_{n-k}^T\}\right\} &= K\sigma_w^2 \text{tr}\left\{E\{\mu_{n-k}^2\}\right. \\ &\quad \left.\sum_{k=0}^{K-1} E\{\mathbf{Q}_{n-k}\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\mathbf{Q}_{n-k}\}\right\} \\ &= K^2\sigma_w^2c^2M\tilde{\sigma}_x^2, \end{aligned} \quad (3.77a)$$

$$\begin{aligned} \text{tr}\left\{\mathbf{R}_{\mathbf{v},n} \sum_{k=0}^{K-1} E\{\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\Gamma_{n-k}^T\}\right\} &= \text{tr}\left\{\mathbf{R}_{\mathbf{v},n} \sum_{k=0}^{K-1} E\{\Gamma_{n-k}^T\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\}\right\} \\ &= \text{tr}\{\mathbf{R}_{\mathbf{v},n}\}Kc\frac{M}{L}\tilde{\sigma}_x^2, \end{aligned} \quad (3.77b)$$

$$\begin{aligned} \text{tr}\left\{E\left\{\sum_{k=0}^{K-1} \Gamma_{n-k}\mathbf{x}_{n-k}\mathbf{x}_{n-k}^T\mathbf{v}_{n-k}\right.\right. \\ \left.\left.\sum_{r=0}^{K-1} \mathbf{v}_{n-r}^T\mathbf{x}_{n-r}\mathbf{x}_{n-r}^T\Gamma_{n-r}^T\right\}\right\} &= \text{tr}\{\mathbf{R}_{\mathbf{v},n}\}c^2(L+2)\frac{M}{L}\tilde{\sigma}_x^2\sigma_x^2K^2. \end{aligned} \quad (3.77c)$$

Substituting (3.77a)-(3.77c) into (3.19), the trace of the system mismatch autocorrelation matrix is then given by

$$\begin{aligned} \text{tr}\{\mathbf{R}_{\mathbf{v},n}\} &= \text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\} + 2(1-\xi)\sigma_s^2L \\ &\quad + K^2\sigma_w^2c^2M\tilde{\sigma}_x^2 - 2\text{tr}\{\mathbf{R}_{\mathbf{v},n-1}\}c\frac{M}{L}\tilde{\sigma}_x^2K \\ &\quad + \text{tr}\{\mathbf{R}_{\mathbf{v},n}\}K^2c^2(L+2)\frac{M}{L}\tilde{\sigma}_x^2\sigma_x^2K^2. \end{aligned} \quad (3.78)$$

As before, assuming that $\mathbf{R}_{\mathbf{v},n}$ is fluctuating around its mean when $n \rightarrow \infty$ and substituting $c = 2\mu/(L\sigma_x^2)$, $\text{tr}\{\mathbf{R}_{\mathbf{v},n}\}$ in (3.78) can be simplified giving the misalignment

$$\begin{aligned} \eta'_{\text{MMax-AP}} &= \frac{K\mu\sigma_w^2}{\sigma_x^2\phi_a} + \frac{L\sigma_x^2(1-\xi)L^2\sigma_s^2}{M\tilde{\sigma}_x^2 2\mu K\phi_a} \\ &= \frac{K\mu\sigma_w^2}{\sigma_x^2\phi_a} + \psi \frac{(1-\xi)L^2\sigma_s^2}{2\mu K\phi_a}, \end{aligned} \quad (3.79)$$

where ϕ_a and ψ are given in (3.46) and (3.62) respectively.

For projection order $K = 1$, the steady-state misalignment performance of MMax-

Table 3.2: Steady-state misalignment for WGN input

Algorithm	η'	ϕ , ϕ_a and β_r
LMS	$\frac{\mu\sigma_w^2 L}{\phi} + \frac{(1-\xi)L\sigma_s^2}{2\mu\sigma_x^2\phi}$	$\phi = 1 - 2\mu\sigma_x^2(1+L/2)$
NLMS	$\frac{\mu\sigma_w^2}{\sigma_x^2\phi} + \frac{(1-\xi)L^2\sigma_s^2}{2\mu\phi}$	$\phi = 1 - (2\mu/L)(1+L/2)$
AP	$\frac{K\mu\sigma_w^2}{\sigma_x^2\phi_a} + \frac{(1-\xi)\sigma_s^2 L^2}{2K\mu\phi_a}$	$\phi_a = 1 - K(2\mu/L)(1+L/2)$
RLS	$\frac{(1-\lambda)L\sigma_w^2}{2\sigma_x^2\phi} + \frac{(1-\xi)L\sigma_s^2}{(1-\lambda)\phi}$	$\phi = 1 - (1-\lambda)(1+L/2)$
MMax-NLMS	$\frac{\mu\sigma_w^2}{\sigma_x^2\phi} + \frac{L\sigma_x^2}{M\tilde{\sigma}_x^2} \frac{(1-\xi)L^2\sigma_s^2}{2\mu\phi}$	$\phi = 1 - (2\mu/L)(1+L/2)$
MMax-AP	$\frac{K\mu\sigma_w^2}{\sigma_x^2\phi_a} + \frac{L\sigma_x^2}{M\tilde{\sigma}_x^2} \frac{(1-\xi)L^2\sigma_s^2}{2\mu K\phi_a}$	$\phi_a = 1 - K(2\mu/L)(1+L/2)$
MMax-RLS	$\frac{(1-\lambda)L^2\sigma_w^2}{\beta_r} + \frac{(1-\xi)2L\sigma_s^2 M\tilde{\sigma}_x^2}{(1-\lambda)\beta_r}$	$\beta_r = 2M\sigma_x^2 - (1-\lambda)L(L+2)\sigma_x^2$

AP is the same as MMax-NLMS as expected. For full adaptation $M = L$, $\tilde{\sigma}_x^2 = \sigma_x^2$ is satisfied and hence MMax-AP is equivalent to AP. In addition, the estimation variance of MMax-AP is independent of the number of taps selected for adaptation M and so, as shown in [24], for stationary unknown system condition $\xi = 1$, the same steady-state misalignment can be achieved with various M . It can be seen that the lag variance is proportional to ψ as defined in (3.62) and hence degradation in steady-state misalignment is expected for reducing M under non-stationary unknown system condition $\xi < 1$.

The steady-state misalignment of various algorithms for non-stationary unknown system conditions are summarized in Table 3.2.

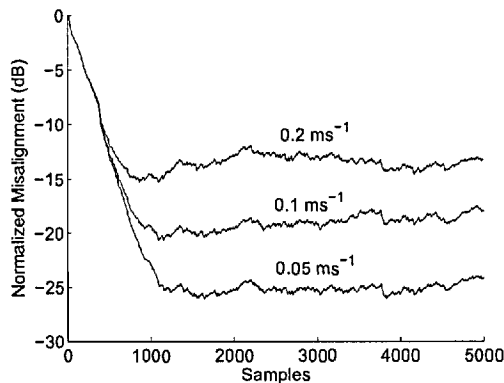


Figure 3.3: NLMS normalized misalignment for various source velocity with impulse response h_n generated using the method of images [$L_R = L = 64$, $\mu = 0.1$, SNR = 40 dB].

3.5 Simulations and results

3.5.1 Comparison between tracking a modified Markov model and the image model

To provide a sense of realistic values for ξ in the modified Markov model given by (3.2) under non-stationary system conditions, a set of impulse responses h_n is first generated using the method of images [72]. In this experimental setup, a microphone is positioned in the centre of a $2.5 \times 3.5 \times 3$ m room. In order to introduce a time-varying h_n of length $L_R = 64$, a zero mean unit variance white Gaussian noise (WGN) source is moved across the receiving room at a velocity of (i) 0.2 ms^{-1} , (ii) 0.1 ms^{-1} and (iii) 0.05 ms^{-1} . An adaptive filter of length $L = 64$, employing the NLMS algorithm with $\mu = 0.1$, is used to track the time-varying impulse response h_n . With reference to Fig. 2.1, an uncorrelated WGN w_n with zero mean is added to the received signal as measurement noise to achieve a signal-to-noise ratio (SNR) of 40 dB. The normalized misalignment η defined in (3.22) is employed to analyze the tracking behaviour of each algorithm. Figure 3.3 shows the normalized misalignment plots corresponding to each moving source's velocity. It can be observed that the steady-state misalignment performance of NLMS increases with reducing source velocity as expected. Figure 3.4 shows the tracking performance of NLMS with impulse response h_n generated using the modified Markov model given by (3.2)

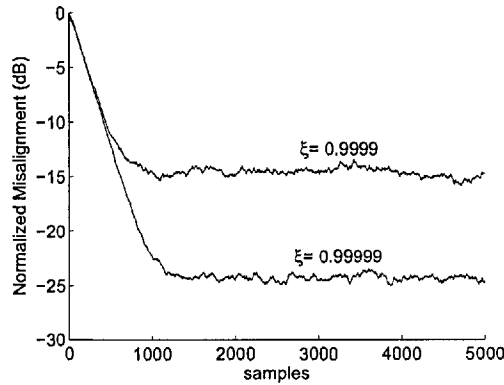


Figure 3.4: NLMS normalized misalignment for various ξ with impulse response \mathbf{h}_n generated using modified Markov model [$L_R = L = 64$, $\mu = 0.1$, $\sigma_s^2 = 1$, SNR = 40 dB].

for time-varying unknown system conditions $\xi = 0.9999$ and $\xi = 0.99999$ with $\sigma_s^2 = 1$. For comparison purposes, the parameters for this modified Markov model experiment are $L_R = L = 64$, $\mu = 0.1$ and SNR = 40 dB as before. Comparing Figs. 3.3 and 3.4, it can be seen that with $\sigma_s^2 = 1$, the tracking performance of NLMS for $\xi = 0.9999$ and $\xi = 0.99999$ defined by (3.2) is comparable to tracking the change in \mathbf{h}_n generated using the method of images with a source moving at a velocity of 0.2 ms^{-1} and 0.05 ms^{-1} respectively. As a consequence, in the following experiments, $\sigma_s^2 = 1$ and values for ξ close to 1 are used to evaluate the tracking performances of algorithms under time-varying unknown system conditions controlled using the modified Markov model.

3.5.2 Effect of non-stationarity on misalignment for NLMS and MMax-NLMS

Simulations to support the analysis of steady-state normalized misalignment for time-varying system identification such as shown in Fig. 2.1 using the modified Markov model is presented in this section. The normalized misalignment η defined in (3.22) is employed. Figure 3.5 shows NLMS results with $\sigma_s^2 = 1$ for a time-invariant system ($\xi = 1$) and three time-varying systems ($\xi = \{0.999999, 0.99999, 0.9999\}$) where smaller values of ξ indicate higher degrees of time-variation as explained in Section 3.2. The input signal is zero mean WGN with $\sigma_x^2 = 1$ and the adaptive filter is of length $L = 64$ while the step-size

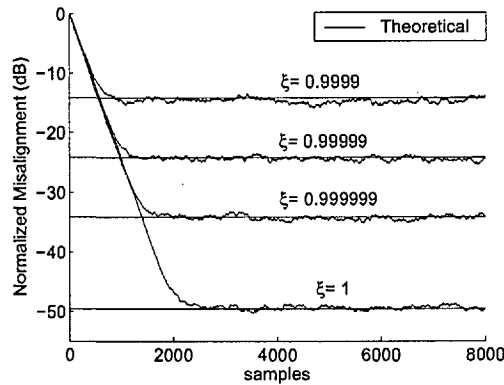


Figure 3.5: NLMS normalized misalignment for various ξ [$L = 64$, $\mu = 0.1$, $\sigma_x^2 = \sigma_s^2 = 1$, SNR = 40 dB].

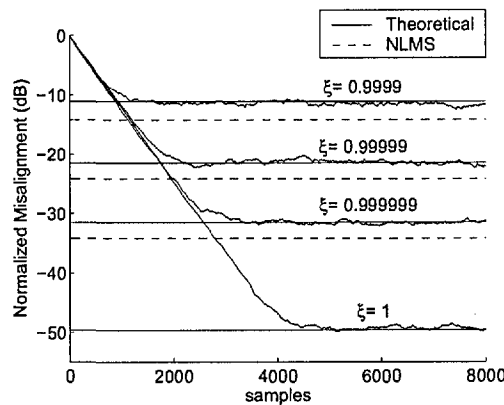


Figure 3.6: MMax-NLMS normalized misalignment for various ξ [$L = 64$, $M = 8$, $\mu = 0.1$, $\sigma_x^2 = \sigma_s^2 = 1$, SNR = 40 dB]. Dashed lines indicate corresponding performance for NLMS.

$\mu = 0.1$ in these examples. The learning curves are averaged over 5 independent trials and theoretical values of η'_{NLMS} given by (3.29) are first normalized with the unknown system $\|\mathbf{h}_n\|_2^2$ before being plotted as superimposed straight horizontal lines. Figure 3.6 shows the results of an equivalent experiment for MMax-NLMS with $L = 64$ and $M = 8$ taps are selected for adaptation at each sample iteration. For comparison purposes, the corresponding theoretical values of η_{NLMS} from the previous experiment are also included in Fig. 3.6 as dashed lines. For both experiments, uncorrelated zero mean WGN sequence w_n is added to achieve a signal-to-noise ratio (SNR) of 40 dB.

The results show that both NLMS and MMax-NLMS are sensitive to time-variation of the unknown system in that the misalignment performance degrades with increasing devi-

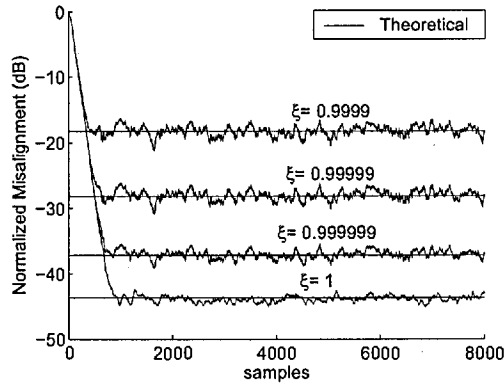


Figure 3.7: AP normalized misalignment for various ξ [$L = 64, K = 3, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40$ dB].

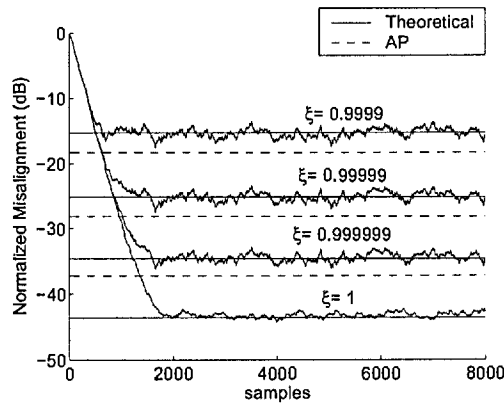


Figure 3.8: MMax-AP normalized misalignment for various ξ [$L = 64, M = 8, K = 3, \mu = 0.1, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40$ dB]. Dashed lines indicate corresponding performance for AP.

ation of ξ from unity. The MMax-NLMS algorithm can be seen to perform approximately 3 to 4 dB worse, in terms of steady-state normalized misalignment, than NLMS under these time-varying conditions. For a time-invariant system $\xi = 1$, both MMax-NLMS and NLMS achieve the same steady-state misalignment since the estimation variance is independent of M as can be seen from (3.29) and (3.60). The MMax-NLMS algorithm, however, has a slower rate of convergence compared to that of NLMS for all cases of $\xi \leq 1$ as expected.

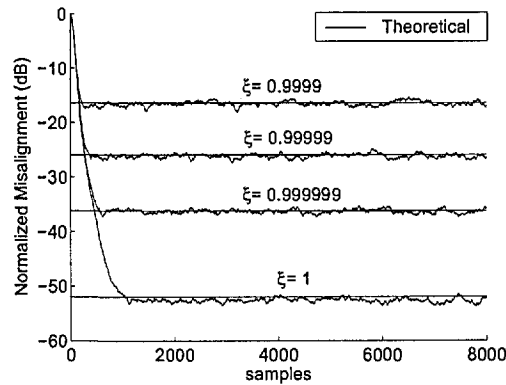


Figure 3.9: RLS normalized misalignment for various ξ [$L = 64, \lambda = 0.9948, \sigma_x^2 = \sigma_s^2 = 1, \text{SNR} = 40$ dB].

3.5.3 Effect of non-stationarity on misalignment for AP and MMax-AP

Figures 3.7 and 3.8 show the normalized misalignment for the AP and MMax-AP respectively where the straight lines indicate theoretical steady-state normalized misalignments for various non-stationary conditions ξ . The dashed lines in Fig. 3.8 represent the theoretical steady-state normalized misalignments for the AP algorithm. In these simulations, the input signal is zero mean WGN with $\sigma_x^2 = 1$ while the adaptive filter is of length $L = 64$ with $\mu = 0.1$. A projection order of $K = 3$ and for MMax-AP, $M = 8$ while uncorrelated zero mean WGN w_n is added to achieve an SNR of 40 dB. In both simulations, the learning curves are averaged over 5 independent trials.

The results indicate that the steady-state normalized misalignment performance degrades for MMax-AP by approximately 2 to 3 dB compared to AP with increasing deviation of ξ from unity. For a time-invariant system, $\xi = 1$, the steady-state normalized misalignment of the MMax-AP algorithm is insensitive to the tap selection since its estimation variance is independent of M as can be seen from (3.79).

3.5.4 Effect of non-stationarity on misalignment for RLS and MMax-RLS

Figures 3.9 and 3.10 show RLS and MMax-RLS normalized misalignment results for various ξ as before. In these simulations, the adaptive filter is of length $L = 64$ and a forgetting

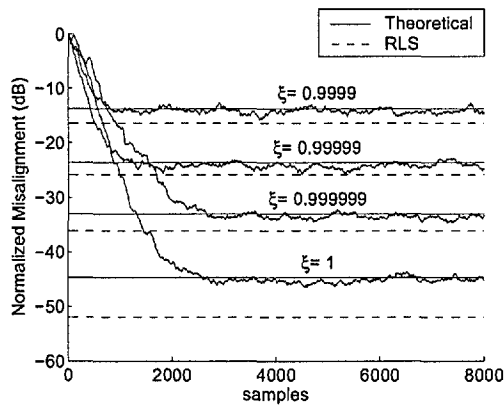


Figure 3.10: MMax-RLS normalized misalignment for various ξ [$L = 64$, $M = 4$, $\lambda = 0.9948$, $\sigma_x^2 = \sigma_s^2 = 1$, SNR = 40 dB]. Dashed lines indicate corresponding performance for RLS.

factor of $\lambda = 1 - 1/(3L) = 0.9948$ is used [39]. As before, the input is a zero mean WGN sequence. With reference to Fig. 2.1 and (3.2), $\sigma_s^2 = \sigma_x^2 = 1$ while an uncorrelated zero mean WGN w_n is added to achieve an SNR of 40 dB for each of the 5 independent trials. For the MMax-RLS algorithm, $M = 4$ taps are selected for adaptation at each iteration.

The results show that both RLS and MMax-RLS are sensitive to time-variation of the unknown system with misalignment performance of MMax-RLS degraded by approximately 7 and 3 dB compared to RLS for $\xi = 1$ and $\xi = 0.9999$ respectively. In contrast to the MMax-NLMS and MMax-AP algorithms, MMax-RLS is sensitive to tap selection even for a non-stationary unknown system $\xi = 1$ since β_r in the estimation variance is dependent on M as shown in (3.73).

3.5.5 Effect of step-size on misalignment for NLMS and MMax-NLMS

Figure 3.11 shows the effect of varying the step-size μ on the steady-state misalignment for NLMS under stationary ($\xi = 1$) and time-varying ($\xi = 0.99999$) unknown system conditions using a zero mean unit variance WGN input sequence. In this experiment, the adaptive filter length is $L = 128$ and an uncorrelated zero mean WGN sequence w_n is added to achieve an SNR of 30 dB. The average steady-state normalized misalignment is obtained from 5 independent trials.

For the stationary case $\xi = 1$, it can be observed that the steady-state normalized misalignment increases with step-size μ as expected. In this simulation example, the mean difference between the experimental and theoretical steady-state normalized misalignment is 0.03 dB. For the case of non-stationary unknown system condition $\xi = 0.99999$, there exists a μ_{mis} giving the lowest misalignment. The theoretical value of $\mu_{\text{mis}} = 0.378$, computed using (3.33), is shown by the vertical dotted line and hence it can be observed that within the region $0 < \mu \leq \mu_{\text{mis}}$, the steady-state misalignment reduces with increasing μ . The mean difference between the experimental and the theoretical normalized misalignment is 0.22 dB.

Figure 3.12 shows the effect of step-size on steady-state misalignment for the MMax-NLMS algorithm under the conditions $\xi = 1$ and $\xi = 0.99999$ with $L = 128$ and $M = 64$. As before, this experiment is simulated using a 30 dB SNR. Similar to the case of NLMS, it can be seen that for $\xi = 1$, the steady-state normalized misalignment increases with step-size μ . For the case of $\xi = 0.99999$, there exists a $\mu_{\text{mis}} = 0.384$ governed by (3.65) which is plotted as a vertical line. The mean differences between experimental and theoretical steady-state normalized misalignment for the cases of $\xi = 1$ and $\xi = 0.99999$ are 0.13 and 0.03 dB respectively. Comparing Figs. 3.11 and 3.12, the normalized misalignment for MMax-NLMS is comparable to that of the NLMS algorithm with this case of $M = 0.5L$ and $\xi = 0.99999$, since as discussed in Section 3.4.1, ψ increases insignificantly within the range $0.5L \leq M < L$.

3.5.6 Effect of tap selection M on normalized misalignment

The effect of tap selection on normalized misalignment with a time-varying unknown system is compared for the MMax-based algorithms using a WGN input sequence with zero mean and unit variance. Figure 3.13 shows the variation of average normalized misalignment with M for MMax-NLMS, MMax-AP and MMax-RLS. The length of the adaptive filter is $L = 128$ while $16 \leq M \leq 128$ and $\xi = 0.9999$. For the MMax-NLMS and MMax-AP algorithms, $\mu = 0.1$ is used while for MMax-RLS a forgetting factor of $\lambda = 1 - 1/(3L) = 0.9896$ is used. The steady-state normalized misalignment for each

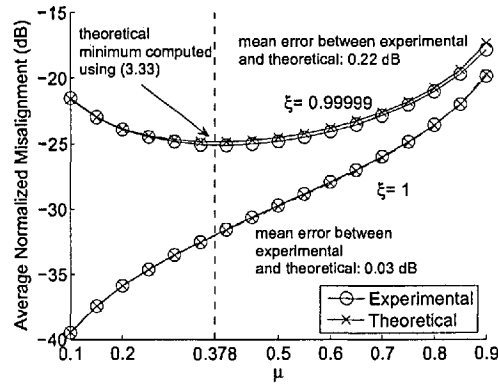


Figure 3.11: NLMS: Variation of average normalized misalignment with μ [$L = 128$, $\sigma_x^2 = \sigma_s^2 = 1$, SNR = 30 dB].

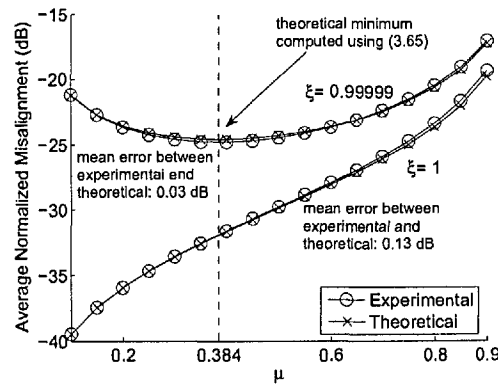


Figure 3.12: MMax-NLMS: Variation of average normalized misalignment with μ [$\sigma_x^2 = \sigma_s^2 = 1$, $L = 128$, $M = 64$, SNR = 30 dB].

algorithm is averaged over 5 independent trials and for each trial, an SNR = 40 dB is achieved by adding an uncorrelated WGN sequence w_n with zero mean to the received microphone signal.

Although the performance of each algorithm is plotted on the same axis, the intention here is not to compare each algorithm's relative normalized misalignment in this simulation example. Instead, it can be seen that for each algorithm, the normalized misalignment reduces with increasing M under the same non-stationary unknown system condition of $\xi = 0.9999$. More importantly, for each algorithm, only a modest degradation in steady-state misalignment performance is observed with reducing M within the range of $0.5L \leq M < L$. When M is reduced further, the degradation in steady-state misalignment

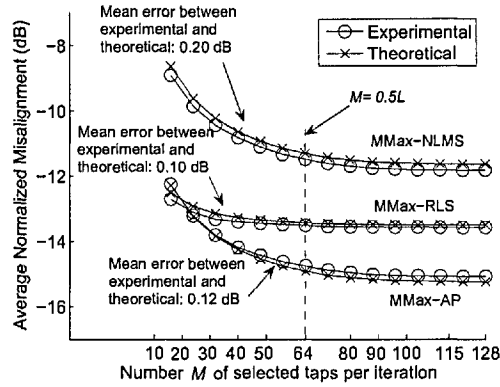


Figure 3.13: Variation of average normalized misalignment with the number of taps selected for adaptation M [$L = 32$, $\mu = 0.1$, $K = 3$, $\lambda = 0.9896$, $\xi = 0.9999$, $\sigma_x^2 = \sigma_s^2 = 1$, SNR = 40 dB].

performance is more pronounced for MMax-NLMS and MMax-AP since, as analyzed and discussed in Fig. 3.2, ψ increases significantly. The degradation in steady-state normalized misalignment performance for MMax-RLS is less pronounced, suffering approximately 1 dB degradation when M is reduced from 128 to 16. The mean errors between theoretical and experimental results in this simulation for MMax-NLMS, MMax-AP and MMax-RLS are 0.20, 0.10 and 0.12 dB respectively. As before, the validity of the analysis presented in this chapter is shown to be valid to within tolerable errors.

3.5.7 Effect of SNR on normalized misalignment

The effect of SNR on steady-state normalized misalignment is investigated for the various MMax selective-tap algorithms under non-stationary unknown system conditions. The experimental parameters for this simulation setup are $L = 128$, $M = 64$, $\xi = 0.99999$, $\mu = 0.1$, projection order $K = 3$ and forgetting factor $\lambda = 1 - 1/(3L) = 0.9974$. The normalized misalignments for each algorithm are averaged over 5 independent trials.

Figure 3.14 shows the variation of MMax-NLMS and MMax-AP normalized misalignments with SNR. For each of the algorithms, the steady-state normalized misalignment improves with increasing SNR as expected. The MMax-AP algorithm is more sensitive to SNR variation achieving an improvement of approximately 9 dB normalized misalignment when SNR is increased from 10 to 40 dB compared to approximately 4 dB for MMax-

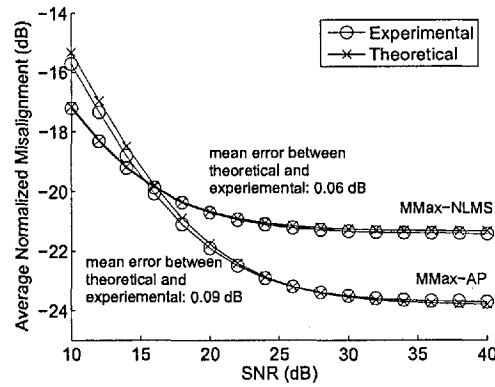


Figure 3.14: MMax-NLMS and MMax-AP: Variation of average normalized misalignment with SNR [$L = 128$, $M = 64$, $\mu = 0.1$, $K = 3$, $\xi = 0.99999$, $\sigma_x^2 = \sigma_s^2 = 1$].

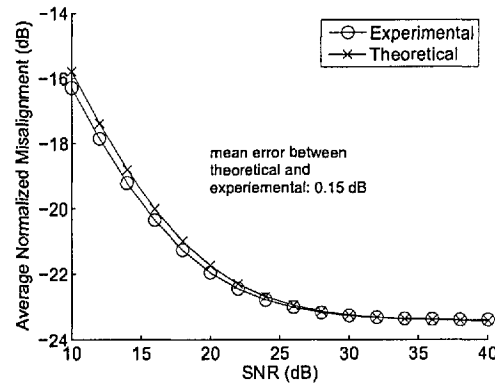


Figure 3.15: MMax-RLS: Variation of average normalized misalignment with SNR [$L = 128$, $M = 64$, $\lambda = 0.9974$, $\xi = 0.99999$, $\sigma_x^2 = \sigma_s^2 = 1$].

NLMS.

Figure 3.15 shows the corresponding normalized misalignment for MMax-RLS under various SNR conditions. As the SNR is increased from 10 to 40 dB, an improvement, though not linearly, of approximately 7.5 dB in steady-state normalized misalignment is observed. The mean error between the theoretical and experimental results are 0.06, 0.09 and 0.15 dB for MMax-NLMS, MMax-AP and MMax-RLS respectively hence verifying the analysis presented in this chapter.

3.6 Discussion and conclusions

Misalignment analysis for a class of MMax-based and fully updated algorithms have been presented which describe their performances when tracking a time-varying unknown system that varies according to a modified first-order Markov model [77]. This analysis can be applied to all algorithms that can be written using the update equation given in (3.9). When the time-variation is removed by setting $\xi = 1$, the analysis yields known results [35] for the various fully updated algorithms. The results for the standard algorithms are consistent with previous work [35] [76] in terms of estimation variance but present new results for the lag variance. Under time-invariant unknown system conditions, the steady-state normalized misalignment for MMax-NLMS and MMax-AP is independent of M while the same is not true for MMax-RLS. For time-varying unknown system conditions, the performance of MMax-based algorithms in terms of steady-state misalignment degrades with increasing time-variation. This degradation is proportional to ψ and as can be seen from Fig. 3.2, the increase in ψ is insignificant with reducing M for $0.5L \leq M < L$ and as a consequence, the degradation in steady-state normalized misalignment performance is negligible. As will be shown in Chapter 4, this property will be exploited for stereophonic AEC algorithms such as presented in [26] [30]. In addition, it has been shown that under time-varying unknown system conditions, there exists for each NLMS and MMax-NLMS, an optimal step-size given by (3.33) and (3.65) respectively, which jointly maximizes the performances in terms of low misalignment and high convergence rate. Due to the nature of stochastic processes and coefficient noise, the theoretical and experimental results are very close but not exactly the same. Nevertheless, simulations presented have been shown to verify the theoretical analysis to within tolerable errors which accurately describes the performances of the algorithms. This analysis enables a judicious trade-off between the computational savings of partial update schemes and their tracking performance.

3.7 Appendix

3.7.1 Fourth-order factorization for zero mean gaussian variables

For an i.i.d. Gaussian distributed signal $x(n)$, the matrix $\Phi = E\{\mathbf{x}_n \mathbf{x}_n^T \mathbf{x}_n \mathbf{x}_n^T\}$ has elements

$$\Phi_{k,l} = E\left\{x(n-k) \sum_{i=1}^L x^2(n-i)x(n-l)\right\},$$

where $\mathbf{x}_n = [x(n) x(n-1) \dots x(n-L+1)]^T$. The factorization property of real zero-mean Gaussian variables [82] is that

$$\begin{aligned} E\{x(i)x(j)x(k)x(l)\} &= E\{x(i)x(j)\}E\{x(k)x(l)\} \\ &\quad + E\{x(i)x(k)\}E\{x(j)x(l)\} \\ &\quad + E\{x(i)x(l)\}E\{x(j)x(k)\}, \end{aligned}$$

from which

$$\begin{aligned} E\{\mathbf{x}_n \mathbf{x}_n^T \mathbf{x}_n \mathbf{x}_n^T\}_{kl} &= 2 \sum_{i=1}^L E\{x(n-k)x(n-i)\}E\{x(n-l)x(n-i)\} \\ &\quad + E\{x(n-k)x(n-l)\} \sum_{i=1}^L E\{x^2(n-i)\}. \end{aligned}$$

From the above, it can be seen that, for the complete matrix, $\Phi = 2\mathbf{R}^2 + \mathbf{R}\text{tr}\{\mathbf{R}\}$.

Now for $x(n)$ i.i.d Gaussian variables

$$E\{x(n-i)x(n-j)\} = \begin{cases} 0, & i \neq j, \\ \sigma_x^2, & i = j, \end{cases}$$

so that $\Phi = (L+2)\sigma_x^4 \mathbf{I}_{L \times L}$.

3.7.2 Verification of step-size boundary condition for MMax-NLMS using contraction mapping

To simplify (3.57), two conditions (i) $\mathcal{T} > -1$ and (ii) $\mathcal{T} < 1$ must be satisfied where

$$\mathcal{T} = \left| 1 - 4\mu \frac{M\tilde{\sigma}_x^2}{L^2\sigma_x^2} + 4\mu^2 \frac{(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} \right|. \quad (3.80)$$

For case (i),

$$\begin{aligned} 1 - 4\mu \frac{M\tilde{\sigma}_x^2}{L^2\sigma_x^2} + 4\mu^2 \frac{(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} &> -1 \\ \frac{2(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} \mu^2 - \frac{2M\tilde{\sigma}_x^2}{L^2\sigma_x^2} \mu + 1 &> 0 \\ \frac{2(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} \left[\mu^2 - \frac{L}{L+2} \mu + \frac{L^3\sigma_x^2}{2(L+2)M\tilde{\sigma}_x^2} \right] &> 0. \end{aligned} \quad (3.81)$$

Introducing the term $\left(L/[2(L+2)] \right)^2$ into the quadratic expression and using the completing the squares approach, (3.81) can be simplified as

$$\frac{2(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} \left[\left(\mu - \frac{L}{2(L+2)} \right)^2 + \frac{L^3\sigma_x^2}{2(L+2)M\tilde{\sigma}_x^2} - \frac{L^2}{4(L+2)^2} \right] > 0 \quad (3.82)$$

Let

$$\psi = L\sigma_x^2/(\tilde{\sigma}_x^2 M), \quad (3.83)$$

from which (3.82) can be simplified as

$$\frac{2(L+2)}{L\psi} \left[\left(\mu - \frac{L}{2(L+2)} \right)^2 + \frac{L\psi}{2(L+2)} - \frac{L^2}{4(L+2)^2} \right] > 0. \quad (3.84)$$

It is shown (c.f. Fig. 3.2) that the term $\psi \geq 1$ when $0 \leq M \leq L$ taps are selected for adaptation. Consequently, $2(L+2)/(L\psi) > 0$ and hence (3.84) can be simplified giving

$$\begin{aligned} \left[\mu - \frac{L}{2(L+2)} \right]^2 + \frac{L\psi}{2(L+2)} - \frac{L^2}{4(L+2)^2} &> 0 \\ \left[\mu - \frac{L}{2(L+2)} \right]^2 + \frac{L^2(2\psi - 1) + 4L\psi}{4(L+2)^2} &> 0 \end{aligned}$$

which is valid $\forall \mu$ since $\psi \geq 1$ and hence case (i) gives trivial solutions for step-size μ .

For case (ii),

$$\begin{aligned} 1 - 4\mu \frac{M\tilde{\sigma}_x^2}{L^2\sigma_x^2} + 4\mu^2 \frac{(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2} &< 1 \\ \frac{(L+2)M\tilde{\sigma}_x^2}{L^3\sigma_x^2}\mu^2 - \frac{M\tilde{\sigma}_x^2}{L^2\sigma_x^2}\mu &< 0. \end{aligned} \quad (3.85)$$

Substituting the definition of ψ from (3.83) into (3.85), the condition

$$\frac{\mu}{\psi} \left[\frac{\mu(L+2)}{L} - 1 \right] < 0 \quad (3.86)$$

is obtained. Under the condition $\mu > 0$ and since $\psi \geq 1$, therefore

$$\begin{aligned} \frac{\mu(L+2)}{L} - 1 &< 0 \\ \mu &< \frac{L}{L+2}, \end{aligned}$$

giving $0 < \mu < L/(L+2)$.

□

Chapter 4

Stereophonic Acoustic Echo Cancellation Employing Tap Selection

*Many things difficult to design
prove easy to perform.*

Samuel Johnson (1709-1784)

4.1 Introduction

STEREOPHONIC tele- and video-conferencing systems have gained much popularity [90] [91] [92] in recent years. In applications such as desktop conferencing and hands-free telephony, stereophonic systems provide telepresence to users by enabling listeners to localize speakers in conference meetings where multiple parties might be conversing at the same time. Similar to single channel acoustic echo cancellation (AEC) as discussed in Chapter 2, the stereophonic acoustic echo canceller (SAEC) such as shown in Fig. 4.1 suppresses the echo returned to the transmission room so as to enable undisturbed communication between the rooms. The disturbance due to echo increases in severity with the propagation delay of the channel.

Unlike the single channel case, a serious problem encountered in SAEC is that the

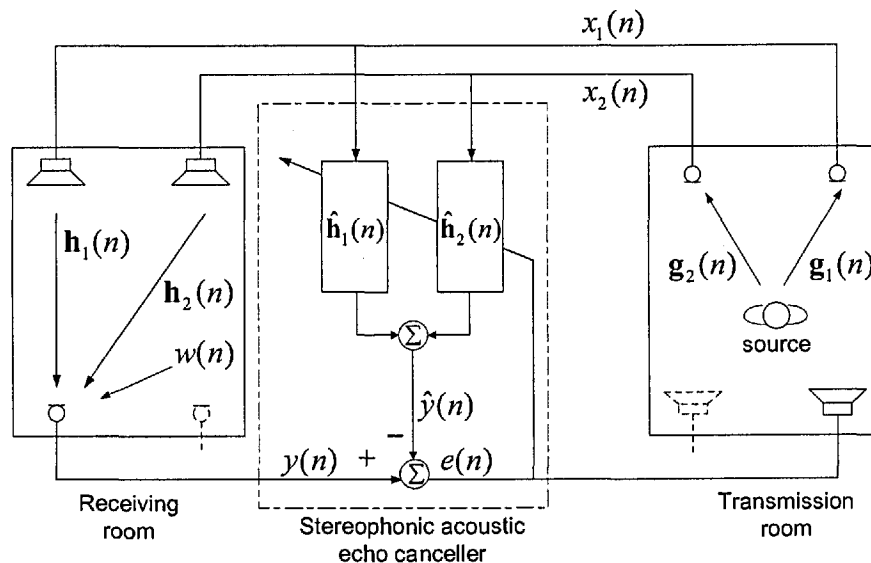


Figure 4.1: Schematic diagram of stereophonic acoustic echo cancellation. Only one channel of the return path is shown for simplicity.

echo canceller coefficients do not in general converge to the true impulse responses of the echo path when the adaptive filters of length L , are greater than or equal in length to those of the transmission room's impulse responses, L_T . In such a situation, solutions for the adaptive filters are non-unique and depend both on the transmission and receiving rooms' impulse responses [52].

In the practical case where $L < L_T$, the problem of non-uniqueness is ameliorated to some degree by the "tail" effect [52]. However, even in such cases, direct application of standard adaptive filtering is not normally successful because the system identification problem is ill-conditioned due to the high interchannel coherence between the two channels' tap-input vectors [52] [90]. This is known as the misalignment problem. As explained in Section 2.2.2, significant undermodelling of the unknown system can also degrade cancellation of echo. To overcome the misalignment problem in this practical case, several approaches have been employed to decorrelate the two input signals using, for example, non-linear processing [11] [52] [93], spectrally shaped random noise [94] [95], comb-filtering [96], leaky extended LMS [97] and alternating fixed-point [98] algorithms. The use of filter updates that are orthogonal to the tap-input vectors has also been considered in [99] [100]. With reference to Fig. 4.1, the common aim of these algorithms is to

achieve decorrelation between input signals $x_1(n)$ and $x_2(n)$ without affecting the quality or stereophonic image of the signals. A survey of existing techniques for SAEC can be found in [101]. In this chapter, in a similar manner to Chapter 2, the dependency of a variable on sample iteration n is shown in brackets while the subscript of a vector quantity is denoted as its channel number. Unless otherwise stated, the first and second subscripts of a scalar quantity are denoted as the channel number and its elemental index of a vector respectively. If only one subscript is shown for the scalar quantity, its elemental index will then be reflected in parenthesis.

In recent years, as discussed in Chapter 2, selective-tap schemes are introduced to reduce computational complexity of, in particular, the NLMS algorithm by updating only a subset of taps at each iteration. The MMax-NLMS algorithm, as discussed in Chapter 2, allows implementation in single channel AEC with performance close to that of the NLMS algorithm. The reduction in computational complexity due to the partial updating is offset to some degree by the computational cost of tap selection which normally requires a sort operation to be performed. However, efficient approximate schemes such as the Short-sort algorithm [71] as discussed in Section 2.7 have been proposed to address this issue. The main motivation of this chapter is *not* the reduction of complexity of SAEC. Instead, tap selection is proposed as a means to reduce interchannel coherence. Drawing on knowledge obtained from developing selective-tap algorithms for single channel AEC as described in Chapter 2, the proposed tap selection algorithm will be applied to normalized least-mean-square (NLMS), affine projection (AP) and recursive least squares (RLS) algorithms for the SAEC application.

This chapter is organized as follows: Problems associated with SAEC are reviewed in Section 4.2. In Section 4.3, it will be shown why direct application of the single channel MMax-NLMS selective-tap algorithm will not achieve sufficient convergence performance for SAEC application. The effect of exclusive tap selection on interchannel coherence and the conditioning of input signal autocorrelation matrix will also be presented. As a proof of concept, an exhaustive tap selection search technique is initially considered in Section 4.4 to demonstrate the selective-tap approach in SAEC. Noting that the exhaus-

tive search technique is not computationally efficient, an efficient exclusive-maximum (XM) tap selection technique involving adaptive filters of realistic order will be presented. This methodology is then applied in combination with a non-linear (NL) processor [52] to form XMNL-based versions of NLMS, AP and RLS in Section 4.5. Computational complexity of the proposed algorithms will also be considered in Section 4.5. Simulation results will be presented in Section 4.6 while Section 4.7 draws conclusions from the work.

4.2 Problems associated with stereophonic acoustic echo cancellation

Stereophonic acoustic echo cancellation (SAEC), as shown in Fig. 4.1, can be viewed as a multichannel extension of the single channel AEC concept. Two microphones are located in the transmission room depicted on the right. The source signal is convolved with the transmission room impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ to give input signals $x_1(n)$ and $x_2(n)$ respectively. These stereophonic signals are then transmitted to loudspeakers in the receiving room which in turn are acoustically coupled to the receiving room microphones. An uncorrelated noise $w(n) = 0$ is initially considered in the development of SAEC algorithms. In addition, only one microphone is considered at the receiving room here for simplicity since similar analysis can be applied to the other channel. Receiving room impulse responses $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$ produce received signal $y(n)$ given by

$$y(n) = \sum_{j=1}^2 \mathbf{h}_j^T(n) \mathbf{x}_j(n), \quad (4.1)$$

where

$$\mathbf{h}_j(n) = [h_{j,0}(n) \ h_{j,1}(n) \ \dots \ h_{j,L_R-1}(n)]^T, \quad j = 1, 2 \quad (4.2)$$

and

$$\mathbf{x}_j(n) = [x_j(n) \ x_j(n-1) \ \dots \ x_j(n-L+1)]^T, \quad j = 1, 2 \quad (4.3)$$

are the j^{th} channel receiving room impulse response of length L_R and tap-input vector of length L respectively while the superscript T denotes transposition operator. Similar to

the single channel case, it is initially assumed that the length of the adaptive filter is the same as that of the unknown impulse response, i.e., $L = L_R$. A pair of adaptive filters each of length L function as a SAEC by estimating the unknown impulse responses $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$ using filter coefficients $\hat{\mathbf{h}}_1(n)$ and $\hat{\mathbf{h}}_2(n)$ where

$$\hat{\mathbf{h}}_j(n) = [\hat{h}_{j,0}(n) \hat{h}_{j,1}(n) \dots \hat{h}_{j,L-1}(n)]^T, \quad j = 1, 2. \quad (4.4)$$

The *a posteriori* error signal between the desired signal and its estimate for this two-channel case is thus given by

$$e_p(n) = y(n) - \sum_{j=1}^2 \hat{\mathbf{h}}_j^T(n) \mathbf{x}_j(n), \quad (4.5)$$

while in a similar manner to the single channel case, the *a priori* error for this stereo case is given by

$$e(n) = y(n) - \sum_{j=1}^2 \hat{\mathbf{h}}_j^T(n-1) \mathbf{x}_j(n). \quad (4.6)$$

Considering the use of the method of least squares following the approach of [52], the time-averaged *a posteriori* cost function can be defined as

$$\mathcal{J}_p(n) = \sum_{i=1}^n \lambda^{n-i} e_p^2(i), \quad (4.7)$$

where $0 \ll \lambda < 1$ is the forgetting factor. Minimizing the least squares criterion (4.7) and using (4.5), the set of normal equations¹

$$\hat{\mathbf{h}}(n) = \Psi^{-1}(n) \Theta(n) \quad (4.8)$$

is obtained, where the two-channel $2L \times 2L$ time-averaged autocorrelation matrix and

¹The rigorous proof of using the *a posteriori* cost function to obtain the normal equations will be shown in Section 5.3.3 in the context of frequency-domain adaptive filtering.

$2L \times 1$ cross-correlation vector are defined [52] respectively as

$$\begin{aligned} \Psi(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^T(i) \\ &= \begin{bmatrix} \Psi_{11}(n) & \Psi_{12}(n) \\ \Psi_{21}(n) & \Psi_{22}(n) \end{bmatrix}_{2L \times 2L} \end{aligned} \quad (4.9)$$

and

$$\Theta(n) = \sum_{i=1}^n \lambda^{n-i} y(i) \mathbf{x}(i), \quad (4.10)$$

given that for this two-channel case,

$$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n) \ \mathbf{x}_2^T(n) \right]^T, \quad (4.11)$$

$$\widehat{\mathbf{h}}(n) = \left[\widehat{\mathbf{h}}_1^T(n) \ \widehat{\mathbf{h}}_2^T(n) \right]^T \quad (4.12)$$

are the $2L \times 1$ concatenated tap-input vectors and filter coefficients respectively.

Defining

$$\mathbf{g}_j(n) = [g_{j,0}(n) \ g_{j,1}(n) \ \dots \ g_{j,L_T-1}(n)]^T \quad (4.13)$$

as the j^{th} channel transmission room impulse response of length L_T , it is shown in [52] and proven in Section 4.8.1, that when $L \geq L_T$, the solution of (4.8) giving a *posteriori* error $e_p(n) = 0$ is in the form

$$\begin{bmatrix} \widehat{\mathbf{h}}_1(n) \\ \widehat{\mathbf{h}}_2(n) \end{bmatrix}_{2L \times 1} = \begin{bmatrix} \mathbf{h}_1(n) \\ \mathbf{h}_2(n) \end{bmatrix}_{2L \times 1} + \varphi(n) \begin{bmatrix} \mathbf{g}_2(n) \\ -\mathbf{g}_1(n) \end{bmatrix}_{2L \times 1}, \quad (4.14)$$

where $\mathbf{g}_j(n)$, for $j = 1, 2$, are appended with $L - L_T$ zeros and $\varphi(n)$ is any scalar quantity. Equation (4.14) indicates that there are non-unique solutions for the adaptive filter coefficients $\widehat{\mathbf{h}}(n)$. More importantly, although there is a mismatch between the estimated and true impulse responses in (4.14), these solutions depend on *both* the transmission and receiving rooms' impulse responses which is undesirable. This is because any changes in the impulse responses of the transmission room, due to for example a change in talker, will require the adaptive filters to reconverge to another solution which again depends

on $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$.

In practical cases where $L < L_T$, the time-averaged autocorrelation matrix $\Psi(n)$ is ill-conditioned because the tap-input vectors $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ are highly correlated². In the case where the adaptive filters are shorter than the lengths of the receiving room impulse responses, L_R , a system mismatch error is introduced in the filter coefficients due to undermodelling. Defining $\check{\mathbf{h}}_j(n)$, of dimension $(L_R - L) \times 1$, as that part of the j^{th} channel in the receiving room impulse response which is not modelled by the respective adaptive filter and $\check{\mathbf{h}}(n) = [\check{\mathbf{h}}_1^T(n) \check{\mathbf{h}}_2^T(n)]^T$, the system mismatch can be quantified [52] by the normalized misalignment $\eta(n)$,

$$\begin{aligned} \eta(n) &= \frac{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}(n)\|_2^2} \\ &= \frac{\check{\mathbf{h}}^T(n) \check{\Psi}^T(n) \Psi^{-2}(n) \check{\Psi}(n) \check{\mathbf{h}}(n)}{\mathbf{h}^T(n) \mathbf{h}(n)}, \end{aligned} \quad (4.15)$$

where $\|\cdot\|_2^2$ is defined as the squared l_2 -norm operator and

$$\begin{aligned} \check{\Psi}(n) &= \sum_{i=1}^n \lambda^{n-i} \begin{bmatrix} \mathbf{x}_1(i) \\ \mathbf{x}_2(i) \end{bmatrix}_{2L \times 1} \begin{bmatrix} \chi_1^T(i) & \chi_2^T(i) \end{bmatrix}_{1 \times 2(L_R-L)} \\ &= \begin{bmatrix} \check{\Psi}_{11}(n) & \check{\Psi}_{12}(n) \\ \check{\Psi}_{21}(n) & \check{\Psi}_{22}(n) \end{bmatrix}_{2L \times 2(L_R-L)} \end{aligned} \quad (4.16)$$

is the time-averaged autocorrelation matrix formed from the “tails” of the input signal $\chi_j(i)$ such that, at each sample iteration i , the $(L_R - L) \times 1$ vector $\chi_j(i)$ is defined as

$$\chi_j(i) = [x_j(i-L) \ x_j(i-L-1) \ \dots \ x_j(i-L_R+1)]^T \quad (4.17)$$

for channels $j = 1, 2$.

The fundamental difference between single channel and stereophonic AEC can be seen from (4.15) and (4.16). In the single channel case, $\check{\Psi}(n)$ in (4.16) reduces to only $\check{\Psi}_{11}(n)$ which consequently reduces the normalized misalignment $\eta(n)$ in (4.15). In addition to

²The relationship between the ill-conditioning of input autocorrelation matrix and interchannel coherence will be derived in the context of frequency-domain variables in Section 5.4.

the error caused by under-modelling of the unknown system, matrices $\check{\Psi}_{12}(n)$ and $\check{\Psi}_{21}(n)$ in (4.16), for the stereo case, are significant because of the high cross-correlation between the two tap-input vectors $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$. As a result, the misalignment for SAEC is significantly higher than for the single channel AEC unless specific processing, such as described in this chapter, is employed.

4.3 Tap selection for SAEC

Partial update schemes achieve complexity reduction by updating only a subset of taps at each iteration. As discussed in Chapter 3, the MMax-NLMS algorithm [21] has been shown to suffer only a modest degradation in convergence rate by updating taps corresponding to the $M = 0.5L$ largest magnitude tap-inputs. It has been shown how ψ , as defined in (3.62) and being inversely proportional to the M-ratio measure \mathcal{M} , affects the steady-state normalized misalignment for the single channel AEC case. In this section, the dependence of convergence rate of MMax-NLMS on \mathcal{M} will be examined and the effect of tap selection on the interchannel coherence for the stereophonic case will be investigated. The measure \mathcal{M} will then be used for the stereo case as an optimization parameter in the subsequent development of the proposed selective-tap SAEC algorithms. For brevity, the discussion in this section will temporarily be limited to MMax-NLMS, although discussions presented here can be generalized to the MMax-AP and MMax-RLS algorithms.

4.3.1 Dependency of convergence rate on \mathcal{M}

In the single channel MMax-NLMS algorithm [21] for an adaptive filter of length L , only those taps corresponding to the M largest magnitude tap-inputs are selected for updating at each iteration. The MMax-NLMS algorithm has been summarized in Table 2.5 of Section 2.10.1.

As can be seen from experimental results presented in Section 2.8, the penalty incurred due to tap selection for the single channel MMax-NLMS algorithm is a decrease in convergence rate for a given step-size μ . The dependency of convergence rate on tap

selection can be examined using the measure \mathcal{M} defined in (3.61) and is reproduced here for convenience,

$$\mathcal{M} = \frac{\|\mathbf{Q}(n)\mathbf{x}(n)\|_2^2}{\|\mathbf{x}(n)\|_2^2}, \quad (4.18)$$

where elements of the diagonal MMax tap selection control matrix

$$\mathbf{Q}(n) = \text{diag}\{[q_0(n) \ q_1(n) \ \dots \ q_{L-1}(n)]\}, \quad (4.19)$$

are defined by (2.47) given as

$$q_i(n) = \begin{cases} 1, & |x(n-i)| \in \{M \text{ maxima of } |\mathbf{x}(n)|\}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.20)$$

for $i = 0, 1, \dots, L-1$ in this single channel case with

$$|\mathbf{x}(n)| = [|x(n)| \ |x(n-1)| \ \dots \ |x(n-L+1)|]^T. \quad (4.21)$$

Whereas the fundamental concept of MMax tap selection was presented in [21], the proposed measure \mathcal{M} provides an explicit quantification of the deviation of the selective-tap case from the full update case such that $\mathcal{M} = 1$ corresponds to full adaptation, i.e., with $M = L$. Furthermore, \mathcal{M} allows direct extension to the stereophonic case as will be shown in Section 4.4.2.

Figure 4.2 shows how \mathcal{M} varies with the size of tap selection M in a single channel case for zero mean, unit variance white Gaussian noise (WGN) tap-input sequence $\mathbf{x}(n)$ at a particular sample iteration n for $L = 256$. It can be seen that \mathcal{M} exhibits only a modest reduction for $0.5L \leq M < L$. Figure 4.3 shows the number of iterations for MMax-NLMS to achieve -20 dB normalized misalignment for various \mathcal{M} and hence verifies the expectation that, over the range $0.5L \leq M < L$, a graceful reduction in convergence rate is obtained as compared to full adaptation [28] [29]. Since convergence rate can be seen to increase monotonically with \mathcal{M} , it is proposed that any degradation in convergence performance due to the subselection of taps can be minimized by selecting taps so as to

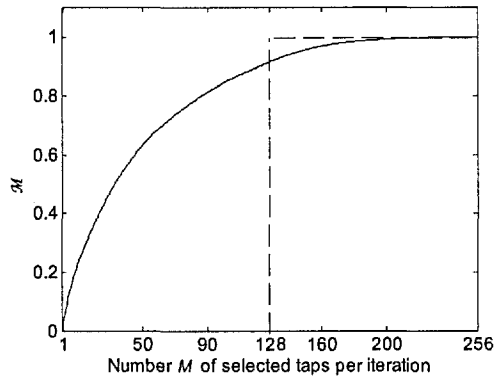


Figure 4.2: Variation of \mathcal{M} with subselection parameter M showing modest reduction of \mathcal{M} within the region $0.5L \leq M < L$ for zero mean unit variance WGN sequence with $L = 256$.

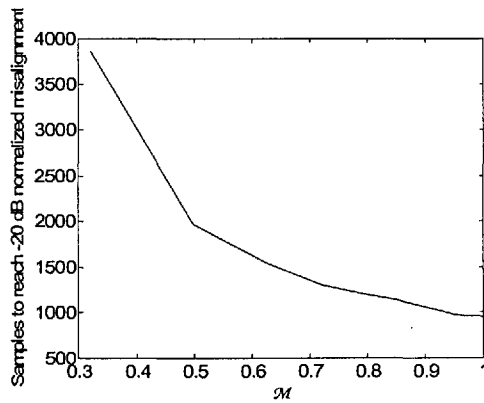


Figure 4.3: Number of iterations for MMax-NLMS to converge to -20 dB normalized misalignment, as a function of \mathcal{M} for $L = 256$.

maximize \mathcal{M} .

4.3.2 Interchannel decorrelation using tap selection

In order to examine the effect of tap selection on interchannel coherence in SAEC, the squared coherence function

$$|\gamma(f)|^2 = \frac{|S_{12}(f)|^2}{S_{11}(f)S_{22}(f)} \quad (4.22)$$

is employed, where $S_{12}(f)$ is the cross power spectrum between the two channels and f is the normalized frequency.

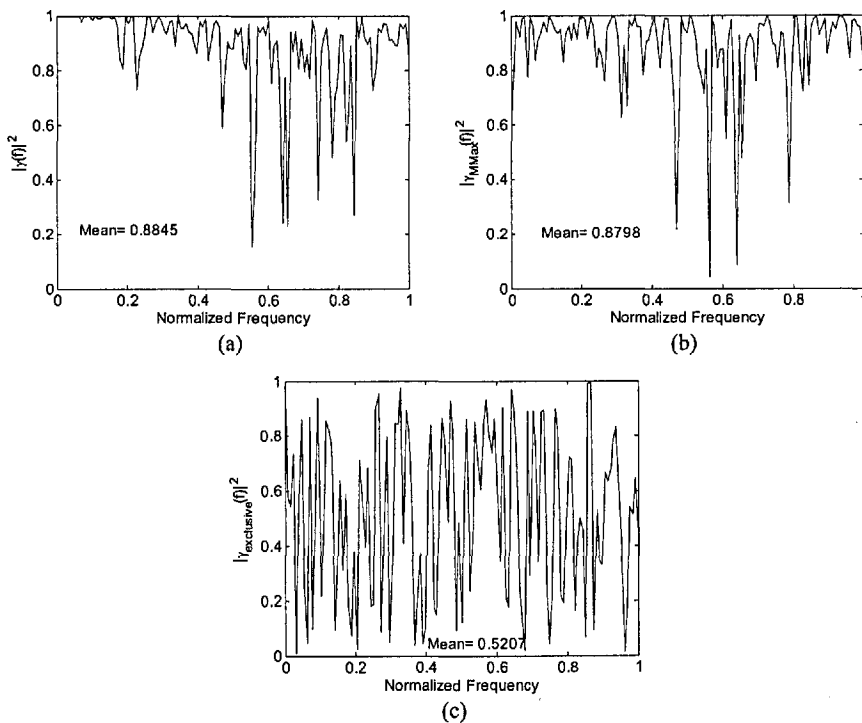


Figure 4.4: Squared interchannel coherence for (a) $M = L = 512$, (b) $M = 0.5L$ with MMax tap selection and (c) $M = 0.5L$ with exclusive tap selection.

As an illustrative example, consider the system of Fig. 4.1 for the case when the source signal is a zero mean unit variance WGN sequence and transmission room impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ are highly correlated each of length 1024. This results in highly correlated tap-input vectors $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ with $\mathbf{x}_j(n) = [x_j(n) \ x_j(n-1) \ \dots \ x_j(n-L+1)]^T$ and for this illustration $L = 512$ is chosen. In this example, transmission room impulse response $\mathbf{g}_1(n)$ is generated using the method of images [72] while $\mathbf{g}_2(n)$ is formed using the following relation

$$\mathbf{g}_2(n) = \epsilon \mathbf{g}_1(n) + (1 - \epsilon) \mathbf{b}(n) , \quad (4.23)$$

where $\mathbf{b}(n)$ is an uncorrelated zero mean unit variance WGN sequence and $0 \leq \epsilon \leq 1$ controls the amount of independent WGN added to $\mathbf{g}_1(n)$. To reflect the high interchannel correlation found in practice, $\epsilon = 0.9$ is used, giving a correlation coefficient of 0.904.

The highly correlated tap-input vectors give rise to a squared coherence close to

one across most of the frequency bands as shown in Fig. 4.4 (a). In the case shown in Fig. 4.4 (b), taps are selected according to the MMax selection criterion with $M = 0.5L$ in both channels. It can be seen clearly that the MMax tap selection does not provide any significant decorrelation. This is because the MMax criterion selects nearly identical tap-indices in both filters, due to the high coherence between the two-channel tap-input vectors. This does not achieve the desired effect of decorrelating the signals.

An exclusive tap selection criterion is considered where selection of the same tap-index in both channels is not permitted. A simple example of such an exclusive case with $M = 0.5L$ (but not the technique used in the proposed algorithm) is to select the taps corresponding to the M largest magnitude tap-inputs in the first channel and the exclusive set of taps in the second channel. Figure 4.4 (c) shows the squared coherence plot of such a case. As can be seen, the interchannel coherence is significantly reduced from a mean of 0.88 to a mean of 0.52 across normalized frequency $0 \leq f \leq 1$, and this is used to illustrate and develop further study of tap selection in Section 4.4.

The exclusive tap selection can be seen as a method for improving the conditioning of the input autocorrelation matrix by considering the case where $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ are highly correlated white Gaussian tap-input vectors. Defining $\mathbf{x}(n) = [\mathbf{x}_1^T(n) \ \mathbf{x}_2^T(n)]^T$ and $E\{\cdot\}$ as the mathematical expectation operator, the two-channel $2L \times 2L$ autocorrelation matrix can be expressed as

$$\begin{aligned} \mathbf{R}_{\mathbf{x}\mathbf{x}} &= E\{\mathbf{x}(n)\mathbf{x}^T(n)\} \\ &= \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix}_{2L \times 2L} \end{aligned} \quad (4.24)$$

Defining $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$ as diagonal exclusive tap selection matrices for channels 1 and 2 respectively, the resulting sparse vectors are given by

$$\tilde{\mathbf{x}}_1(n) = \mathbf{Q}_1(n)\mathbf{x}_1(n), \quad (4.25a)$$

$$\tilde{\mathbf{x}}_2(n) = \mathbf{Q}_2(n)\mathbf{x}_2(n) \quad (4.25b)$$

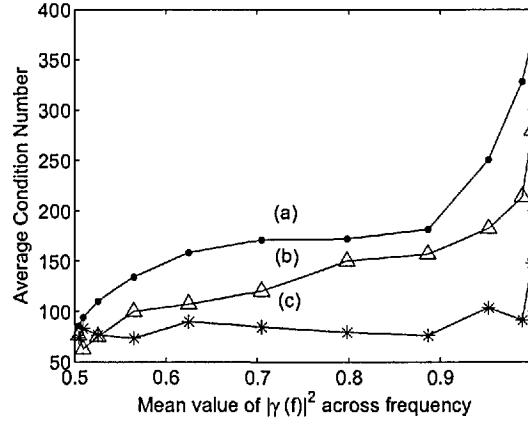


Figure 4.5: Effect of exclusive tap selection for various mean value of $|\gamma(f)|^2$ across frequency on mean condition number for zero mean unit variance WGN sequence (a) without tap selection, (b) with MMax tap selection and (c) with exclusive tap selection.

which in turn give rise to $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ where the diagonals and some off-diagonal elements of \mathbf{R}_{12} and \mathbf{R}_{21} are now zero since based on the definition of exclusive tap selection,

$$\mathbf{Q}_1(n)\mathbf{Q}_2(n) = \mathbf{0}_{L \times L}, \quad (4.26)$$

where $\mathbf{0}_{L \times L}$ is defined as a $L \times L$ null matrix. This improves on the conditioning of $\mathbf{R}_{\mathbf{xx}}$ and in the limit where the subselected tap-input vectors $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$ are perfectly uncorrelated, the autocorrelation matrix is a diagonal matrix given by

$$\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}} = \text{diag}\{\tilde{\sigma}_1^2 \dots \tilde{\sigma}_1^2 \quad \tilde{\sigma}_2^2 \dots \tilde{\sigma}_2^2\} \quad (4.27)$$

with an l_2 condition number of

$$\|\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}\|_2 \|\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}\|_2 = \frac{\max(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2)}{\min(\tilde{\sigma}_1^2, \tilde{\sigma}_2^2)}, \quad (4.28)$$

where $\tilde{\sigma}_j^2$ is the j^{th} channel subselected tap-input variance given by (3.48).

Figure 4.5 shows the variation of mean condition number of time-averaged autocorrelation matrices $\mathbf{R}_{\mathbf{xx}}$ and $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ as a function of the mean of $|\gamma(f)|^2$ across $0 \leq f \leq 1$ where $|\gamma(f)|^2$ is defined in (4.22). The autocorrelation matrices are formed from tap-input

vectors $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ generated by the convolution of a zero mean unit variance WGN source sequence with transmission room impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ generated using the method of images [72], such that $|\gamma(f)|^2$ is controlled by ϵ as described in (4.23) while the additional MMax or exclusive tap selection criterion has been imposed when generating $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$. For each case of mean coherence, the average condition number for 30 trials is computed and plotted as shown in Fig. 4.5 (a), (b) and (c) for $\mathbf{R}_{\mathbf{x}\mathbf{x}}$, MMax $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ and exclusive $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ respectively. It can be observed that as the mean interchannel coherence reduces, $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ become less correlated and hence a reduction of the mean condition numbers for the autocorrelation matrices are exhibited. The explicit link between interchannel coherence and condition number of $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ will be presented in Chapter 5 in the context of frequency-domain quantities. With reference to Fig. 4.5, for each case of mean interchannel coherence, $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ formed from exclusive tap selection has the lowest mean condition number compared to $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ and $\mathbf{R}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}$ formed from MMax tap selection. Hence the exclusive tap selection gives rise to a better conditioned autocorrelation matrix which in turn allows one to address the misalignment problem caused by the ill-conditioned input autocorrelation matrix as discussed in Section 4.2.

4.4 Exclusive-maximum tap selection

4.4.1 Formulation

It has been shown through a simulation example in Section 4.3.2 that exclusive tap selection can improve the conditioning of the two-channel input autocorrelation matrix in SAEC. An adaptive filtering scheme which makes use of this concept without degrading convergence due to partial adaptation is developed in this section. The problem can be formulated as a joint optimization of maximizing the MMax criterion, determined by the M-ratio \mathcal{M} , and minimizing the interchannel coherence under the control of tap selection. This is done using two variables: magnitude weighting, $0 \leq \vartheta_m \leq 1$, to describe the “closeness” of the tap selection to that of the MMax scheme, and coherence weighting, $\vartheta_c = 1 - \vartheta_m$, to describe interchannel coherence between the subsampled tap-input

k,r	Channel Tap-index			
	0	1	2	3
1				
2				
3				
4				
5				
6				


Selected taps


Figure 4.6: Possible combinations of tap selection for $L = 4$ and $M = 2$ in channels 1 and 2.

vectors. A magnitude weighting of $\vartheta_m = 1$ corresponds to selecting coefficients based on the MMax tap selection criterion only. In the following, the dependence of variables on sample iteration n is temporarily omitted for clarity of notation.

Consider firstly the ${}^L C_M$ possible combinations of selecting $M = 0.5L$ taps from each channel's adaptive filter of length L . Let the combinations be indexed $k, r = 1, 2, \dots, {}^L C_M$ giving tap selection sets $\{\zeta_k\}$ and $\{\zeta_r\}$ for channels 1 and 2 respectively and define $\{\zeta_{kr}\}$ as the combined two-channel tap selection set for each sample iteration while $\tilde{\mathbf{x}}_{1,k}$ and $\tilde{\mathbf{x}}_{2,r}$ are the subselected input vectors using tap selection sets $\{\zeta_k\}$ and $\{\zeta_r\}$ respectively. Figure 4.6 shows the different possible combinations of selecting the filter coefficients in each channel for an example case of $L = 4$ and $M = 2$. Clearly, there are ${}^L C_M = 6$ possible combinations of tap selection for each of the two channels.

For the structure shown in Fig. 4.1, \mathbf{A} and \mathbf{C} are defined, for each sample iteration, as square matrices each with dimension ${}^L C_M \times {}^L C_M$ such that the $\{k^{\text{th}}, r^{\text{th}}\}$ element of each matrix contains

$$a_{kr} = \left\| |\tilde{\mathbf{x}}_{1,k}| + |\tilde{\mathbf{x}}_{2,r}| \right\|_1, \quad (4.29)$$

$$c_{kr} = \left\langle \frac{|S_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_r}(f)|^2}{S_{\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_k}(f) S_{\tilde{\mathbf{x}}_r \tilde{\mathbf{x}}_r}(f)} \right\rangle \quad (4.30)$$

respectively where a_{kr} denotes the absolute sum of the selected tap-inputs in a particular tap selection set ζ_{kr} and c_{kr} is the squared coherence, with $\langle \cdot \rangle$ indicating averaging over frequency, of the two tap-input vectors with $L - M$ unselected inputs in each channel set to zero.

\mathcal{A}		ζ_r					
		1	2	3	4	5	6
ζ_k	1	12	34	35	5	22	32
	2	10	4	11	14	19	21
	3	9	33	27	31	3	6
	4	30	17	20	29	18	15
	5	1	16	28	2	7	24
	6	25	26	13	36	23	8

\mathcal{C}		ζ_r					
		1	2	3	4	5	6
ζ_k	1	22	24	25	19	10	1
	2	7	4	21	6	1	9
	3	3	12	23	1	2	8
	4	13	14	1	27	30	31
	5	5	1	15	26	28	29
	6	1	11	20	16	18	17

Figure 4.7: Cost matrices for \mathcal{A} and \mathcal{C} .

Since elements in matrix \mathbf{A} contain the sums of magnitudes which are required to be maximized, an integer cost is first associated with each of the elements a_{kr} such that the *least cost* is allocated to the element having the *largest magnitude* in \mathbf{A} . This new magnitude cost matrix is now defined as \mathcal{A} . In a similar manner, each element in \mathbf{C} will be allocated an integer cost such that element corresponding to the *minimum squared coherence* is allocated the *least cost*. This new coherence cost matrix is then defined as \mathcal{C} . Hence matrices \mathcal{A} and \mathcal{C} contain integer cost values, at each sample iteration, depending on the magnitude sum and interchannel coherence.

To illustrate the above description, Fig. 4.7 shows an example of matrices \mathcal{A} and \mathcal{C} at one particular sample iteration for the case of $L = 4$ and $M = 2$ as before. It can be seen that ζ_{51} gives the lowest cost in \mathcal{A} and consequently, from Fig. 4.6, $|x_{1,0}(n)| + |x_{1,2}(n)| + |x_{2,2}(n)| + |x_{2,3}(n)|$ achieves the highest magnitude amongst all the possible combination sets $k, r = 1, 2, \dots, {}^L C_M$ where $x_{j,i}(n)$ is defined as the i^{th} tap-index corresponding to the j^{th} channel. It is interesting to note that elements in the skew (top-right-to-lower-left) diagonal of \mathcal{C} , as shown in Fig. 4.7, have the lowest cost. This is because these combinations $\{\zeta_{61}, \zeta_{52}, \zeta_{43}, \zeta_{34}, \zeta_{25}, \zeta_{16}\}$ correspond to the exclusive tap selection of the two-channels as can be seen from Fig. 4.6.

A total cost matrix \mathcal{V} is then given, at each sample iteration, by

$$\mathcal{V} = \vartheta_m \mathcal{A} + \vartheta_c \mathcal{C} . \quad (4.31)$$

Defining $\{\zeta_{\min}\} = \{\zeta_{k_{\min}, r_{\min}}\}$ as the tap selection set having minimum cost in matrix \mathcal{V} ,

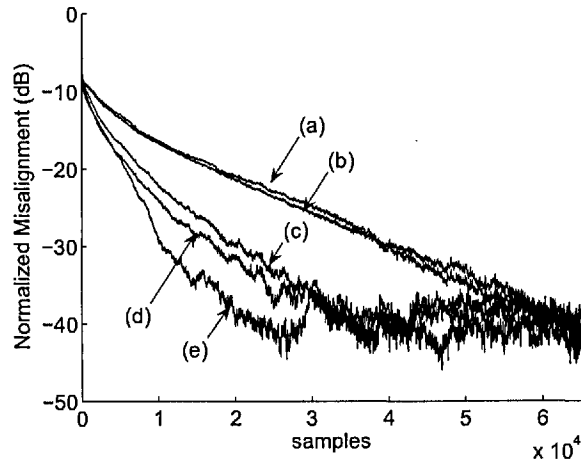


Figure 4.8: Misalignment for (a) $\vartheta_m = 1$, (b) NLMS, (c) $\vartheta_m = 0.9$, (d) $\vartheta_m = 0.7$, (e) $\vartheta_m = 0.1$ [$L = 6$, $M = 3$, $\mu = 0.6$, $\epsilon = 0.9$, $\text{SNR} = 40$ dB].

$\{\zeta_{\min}\}$ can be obtained using

$$k_{\min}, r_{\min} = \arg \min_{k,r} (\mathbf{V}), \quad k, r = 1, 2, \dots, L C_M. \quad (4.32)$$

For small L and letting $\hat{\mathbf{h}}(n) = [\hat{\mathbf{h}}_1^T(n) \hat{\mathbf{h}}_2^T(n)]^T$ and $\mathbf{x}(n) = [\mathbf{x}_1^T(n) \mathbf{x}_2^T(n)]^T$, \mathbf{V} can be searched exhaustively, for each iteration n , so that the tap selection set ζ_{\min} can then be incorporated into NLMS adaptation [26] as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{Q}(n) \frac{2\mu \mathbf{x}(n) e(n)}{\|\mathbf{x}(n)\|_2^2 + \delta_{\text{NLMS}}}, \quad (4.33)$$

with

$$\mathbf{Q}(n) = \text{diag}\left\{[\mathbf{q}_1^T(n) \mathbf{q}_2^T(n)]\right\} \quad (4.34)$$

being the two-channel tap selection control matrix where, at each sample iteration n , element u of $\mathbf{q}_1(n)$ and element v of $\mathbf{q}_2(n)$ are defined for $u, v = 0, 1, \dots, L-1$ as

$$\{q_{1,u}(n), q_{2,v}(n)\} = \begin{cases} 1, & \text{if } u, v \in \{\zeta_{\min}\}, \\ 0, & \text{otherwise.} \end{cases}$$

Figure 4.8 shows simulation results for the normalized misalignment with different

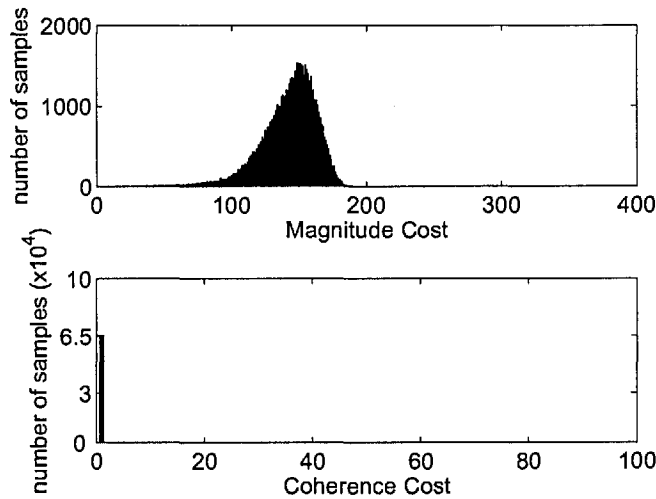


Figure 4.9: Number of samples that a particular integer cost in matrices (a) \mathcal{A} and (b) \mathcal{C} associated with ζ_{\min} is selected for adaptation [$L = 6$, $M = 3$, $\vartheta_m = 0.1$, $\mu = 0.6$, $\epsilon = 0.9$, SNR = 40 dB].

values of magnitude weighting ($\vartheta_m = 0.1, 0.7, 0.9, 1.0$). In this example the source is a zero mean unit variance WGN sequence with adaptive filters having 6 taps per channel and for every iteration, 3 taps are updated ($L = 6$, $M = 3$) using an arbitrarily chosen step-size of $\mu = 0.6$. The relationship between transmission room impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ with lengths $L_T = 12$ is again determined by (4.23) with $\epsilon = 0.9$. The receiving room impulse responses $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$ are taken from a zero mean unit variance WGN sequence and are of lengths $L_R = 6$. This choice of L_T and L_R allows one to study the adaptive filters which uniquely determine the unknown system whilst minimizing the normalized misalignment caused by undermodelling. The normalized misalignment for only one of the two channels is plotted for each case of ϑ_m for reasons of clarity. With reference to Fig. 4.1, an uncorrelated zero mean WGN sequence $w(n)$ is added to achieve a signal-to-noise ratio (SNR) of 40 dB. The simulation result shows that $\vartheta_m = 1$ coincides with MMax-NLMS where performance is close to that of the fully updated NLMS as expected. The highest convergence rate can be seen when $\vartheta_m = 0.1$ (i.e., $\vartheta_c = 0.9$) where a high weighting is given to minimization of the interchannel coherence.

Figure 4.9 shows, for the same experimental setup with $\vartheta_m = 0.1$, the number of iterations that a particular integer cost in matrices (a) \mathcal{A} and (b) \mathcal{C} associated with

ζ_{\min} is selected for adaptation. The highest integer cost for \mathcal{A} and \mathcal{C} in this example is $({}^6C_3)^2 = 400$ and $({}^6C_3)^2 - ({}^6C_3 - 1) = 381$ respectively³. The first 100 integer costs in \mathcal{C} are shown here as there are no tap selection set corresponding to other costs for any of the iterations. It can be seen from Fig. 4.9 (b), that tap combinations corresponding to the lowest cost in \mathcal{C} are being selected for updating throughout the adaptation process from $n = 1$ to $n = 6.5 \times 10^4$ since, for $\vartheta_m = 0.1$, a high weighting is given to the minimization of \mathcal{C} . Upon further investigation, it was found that for $\vartheta_m = 0.1$, all the tap selection sets used for adaptation maximizes a_{kr} , with $k, r = 1, 2, \dots, {}^L C_M$, subject to the exclusive criterion such that combinations k and r contain no tap-indices in common, i.e.,

$$\zeta_k \cap \zeta_r = \{ \}, \quad \forall n, \quad (4.35)$$

where $\{ \}$ is defined as a null set. Therefore the optimization problem can now be redefined in a simpler form of a search where \mathcal{M} is maximized at each sample iteration subject to the constraint given by (4.35).

4.4.2 Efficient realization: the exclusive-maximum tap selection

Since the total cost matrix \mathcal{V} is of dimension ${}^L C_M \times {}^L C_M$, an exhaustive search of \mathcal{V} for the optimum tap selection is computationally expensive for adaptive filters of higher orders and as a consequence, an efficient tap selection scheme is proposed. As mentioned, the objective is to develop a tap selection algorithm which maximizes \mathcal{M} jointly for both channels subject to an exclusivity constraint. In this subsection, the dependence of variables on sample iteration n is temporarily omitted for brevity.

Let

$$\mathbf{p} = |\mathbf{x}_1| - |\mathbf{x}_2| \quad (4.36)$$

be the interchannel tap-input magnitude difference vector and

$$\check{\mathbf{p}} = [\check{p}_0 \check{p}_1 \dots \check{p}_{L-1}]^T, \quad (4.37)$$

³The highest integer cost in \mathcal{C} is always less than that of \mathcal{A} since, as can be seen from Fig. 4.7, elements in the skew diagonal of \mathcal{C} have the same cost integer of 1.

where

$$\check{p}_0 > \check{p}_1 > \dots > \check{p}_{L-1} \quad (4.38)$$

are defined as \mathbf{p} sorted in descending order. Let $\check{x}_{1,i}$ and $\check{x}_{2,i}$ denote the i^{th} tap-input samples of channel 1 and 2, ordered according to the sorting of $\check{\mathbf{p}}$ such that

$$\check{p}_i = |\check{x}_{1,i}| - |\check{x}_{2,i}|, \quad (4.39)$$

for $i = 0, 1, \dots, L-1$. In this two-channel case, the M-ratio \mathcal{M} is defined as

$$\mathcal{M} = \frac{\|\mathbf{Q}\mathbf{x}\|^2}{\|\mathbf{x}\|^2} \quad (4.40)$$

with $\mathbf{Q} = \text{diag}\{\mathbf{q}_1^T \mathbf{q}_2^T\}$ being the two-channel concatenated tap selection matrix and $\mathbf{x} = [\mathbf{x}_1^T \mathbf{x}_2^T]^T$. It has been analyzed in Sections 3.4.1 and 4.3.1 that the NLMS algorithm suffers from insignificant degradation in terms of steady-state misalignment and convergence rate for $0.5L \leq M < L$. As a consequence, $M = 0.5L$ is considered which, in addition, allows the proposed tap selection to satisfy the exclusivity criterion.

As verified in Section 4.8.2, the *exclusive* tap selection set that *maximizes* \mathcal{M} jointly for both channels contains the M largest elements of \mathbf{p} from channel 1 and the M smallest elements of \mathbf{p} from channel 2, i.e.,

$$\{\check{x}_{1,0} \dots \check{x}_{1,M-1} \check{x}_{2,M} \dots \check{x}_{2,L-1}\}. \quad (4.41)$$

Hence at each iteration, assuming no channel gain mismatch between the two channels, the *exclusive-maximum* (XM) tap selection is defined by element u of \mathbf{q}_1 and element v of \mathbf{q}_2 such that for $u, v = 0, 1, \dots, L-1$ and $M = 0.5L$,

$$q_{1,u} = \begin{cases} 1, & p_u \in \{M \text{ maxima of } \mathbf{p}\}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.42a)$$

$$q_{2,v} = \begin{cases} 1, & p_v \in \{M \text{ minima of } \mathbf{p}\}, \\ 0, & \text{otherwise.} \end{cases} \quad (4.42b)$$

As an illustration, consider an SAEC system with channels $j = 1, 2$, adaptive filters each of length $L = 4$ with tap-input vector

$$\mathbf{x}_j(n) = [x_{j,0}(n) \ x_{j,1}(n) \ x_{j,2}(n) \ x_{j,3}(n)]^T. \quad (4.43)$$

The tap-input magnitude difference vector \mathbf{p} may then be expressed as

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} |x_{1,0}(n)| \\ |x_{1,1}(n)| \\ |x_{1,2}(n)| \\ |x_{1,3}(n)| \end{bmatrix} - \begin{bmatrix} |x_{2,0}(n)| \\ |x_{2,1}(n)| \\ |x_{2,2}(n)| \\ |x_{2,3}(n)| \end{bmatrix}. \quad (4.44)$$

Consider the example case $p_2 > p_1 > p_0 > p_3$, for a particular sample iteration. Since $p_2 + p_1 > \dots > p_0 + p_3$, it can be shown that

$$\begin{aligned} |x_{1,2}(n)| - |x_{2,2}(n)| + |x_{1,1}(n)| - |x_{2,1}(n)| &> \dots > |x_{1,0}(n)| - |x_{2,0}(n)| \\ &+ |x_{1,3}(n)| - |x_{2,3}(n)| \\ |x_{1,2}(n)| + |x_{1,1}(n)| + |x_{2,0}(n)| + |x_{2,3}(n)| &> \dots > |x_{1,0}(n)| + |x_{1,3}(n)| \\ &+ |x_{2,1}(n)| + |x_{2,2}(n)|, \end{aligned} \quad (4.45)$$

where \dots refers to all other pair-wise combinations of p_0 , p_1 , p_2 and p_3 . Thus taps corresponding to inputs $x_{1,2}(n)$, $x_{1,1}(n)$, $x_{2,0}(n)$ and $x_{2,3}(n)$ maximize \mathcal{M} with the minimum coherence constraint satisfied by the exclusivity of the tap selection at each sample iteration.

In this way, the XM tap selection criterion efficiently selects the best exclusive sets of taps where best here is defined as nearest to MMax jointly for both channels in order to minimize the degradation in convergence performance due to tap selection. This is achieved by maximizing the \mathcal{M} measure computed using the taps from both channels. Because of the exclusivity constraint, neither channel in general attains a tap selection as good as MMax and some degradation in convergence performance is therefore to be expected. Nevertheless, results presented in Section 4.6 indicate that such degradation is

small compared to the improvement in convergence due to the decorrelating property of XM tap selection.

Note that the XM tap selection criterion as described above will result in a selected tap-input vector with lower power than for the MMax criterion for each channel due to the exclusivity constraint. It is to be expected therefore that the effect of noise may be relatively more significant in the proposed scheme compared to the MMax scheme. However simulation results indicate that any such effects are insignificant compared to the improvements obtained due to the decorrelating properties of the proposed tap selection.

As a final comment, it is irrelevant to consider other exclusive tap selection sets given in the skew diagonal of \mathbf{C} since they have smaller magnitude sum. This approach allows the XM algorithm to eliminate ${}^L C_M \times {}^L C_M - 1$ possible combinations thus allowing efficient implementation of the XM tap selection. Such efficient practical schemes will be developed in Section 4.5 for use with NLMS, AP and RLS adaptation.

4.5 Exclusive-maximum adaptive filtering

As has been shown in Section 4.3.2, the XM tap selection can improve conditioning of the input autocorrelation matrix \mathbf{R}_{xx} and hence improved convergence is expected. The effect of tap selection for the AP and RLS cases on the autocorrelation matrix will be seen to be similar to that which occurs in the NLMS case shown in Section 4.3.2. The XM approach relies on the existence of a unique solution for the adaptive filter coefficients. As will be shown through simulations in Section 4.6, XM tap selection in combination with a non-linear (NL) preprocessor [52] leads to better performance than the use of the NL-preprocessor alone. This combination of XM and NL approach, which will be referred to as XMNL, is highly effective for the cases considered and therefore this combined structure will be proposed for the later experiments. Figure 4.10 shows the schematic diagram of the proposed XMNL-based SAEC structure.

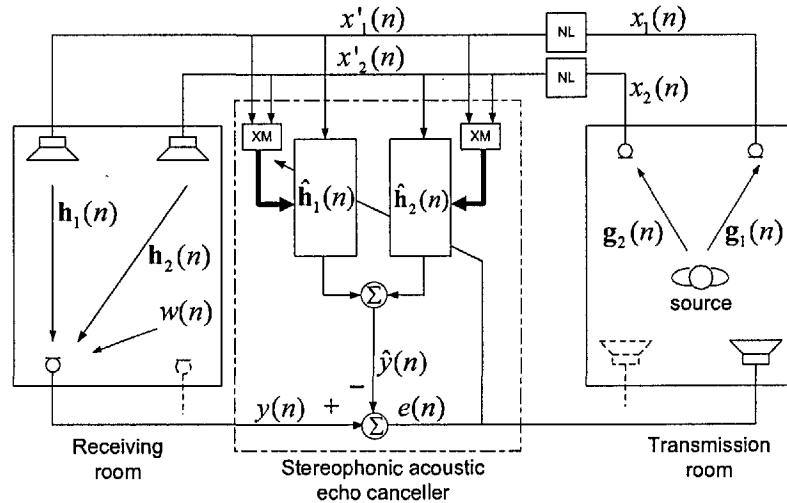


Figure 4.10: Schematic diagram of the XMNL preprocessor in stereophonic acoustic echo canceller. Bold arrows indicate tap selection control.

4.5.1 The XM-NLMS algorithm

The XM tap selection technique can be incorporated into NLMS by selecting taps corresponding to $M = 0.5L$ largest elements of the input magnitude difference vector $\mathbf{p}(n)$ in the first channel and the M smallest elements of $\mathbf{p}(n)$ in the second channel as shown in (4.42a) and (4.42b). Taps are then updated using (4.33), (4.42a) and (4.42b). Simulation result for the XM-NLMS algorithm is shown in Fig. 4.13.

4.5.2 The XMNL-NLMS algorithm

The non-linear (NL) preprocessor [52] implemented using the half-wave rectifier is one of the most effective methods of achieving signal decorrelation without significantly affecting stereo perception. Using $0 < \beta \leq 0.5$ as the non-linearity constant, the input signals

$$\mathbf{x}'_1(n) = \mathbf{x}_1(n) + 0.5\beta[\mathbf{x}_1(n) + |\mathbf{x}_1(n)|], \quad (4.46a)$$

$$\mathbf{x}'_2(n) = \mathbf{x}_2(n) + 0.5\beta[\mathbf{x}_2(n) - |\mathbf{x}_2(n)|], \quad (4.46b)$$

are obtained. Several alternative types of non-linearity techniques such as those reported in [102] can be employed in SAEC for reducing the interchannel coherence. It

has been found however that the non-linearity achieved using the half-wave rectifier, defined by (4.46a) and (4.46b), is the simplest to implement and only minimally affects the speech quality [103] though for music signals, relatively low values of β must be maintained to achieve good perceptual quality. It has also been reported in [11] that the level of distortion measured using the Itakura-Saito measure [104] is small with $\beta = 0.5$. The use of the NL preprocessor (using the half-wave rectifier) with NLMS adaptation will be referred to as NL-NLMS. Several workers [45] [92] [96] have proposed algorithms in combination with the NL processor so as to achieve good convergence performance. In the same manner, a combined algorithm will be proposed employing XM tap selection in addition to the NL preprocessor so as to improve the convergence rate obtained from the use of the NL preprocessor alone. It has been shown [27] that the resulting XMNL-NLMS algorithm can give useful levels of convergence with performance close to that of an existing RLS-based approach [52]. The XMNL-NLMS algorithm is summarized in Table 4.2 of Section 4.8.3.

4.5.3 The XMNL-AP algorithm

The affine projection (AP) algorithm [35] incorporates multiple projections by concatenating past input vectors from sample iteration n to $n - K + 1$ where K is defined as the projection order. Similar to the single channel MMax-AP algorithm discussed in Section 2.6.1, the concatenated *subselected* tap-input vector and the concatenated *full* tap-input vector can be distinguished by first letting

$$\tilde{\mathbf{x}}'(n) = \mathbf{Q}(n)\mathbf{x}'(n) \quad (4.47)$$

be the two-channel subselected tap-input vector where $\mathbf{x}'(n) = [\mathbf{x}_1^T(n) \ \mathbf{x}_2^T(n)]^T$ is the concatenated NL-processed $2L \times 1$ tap-input vector and $\mathbf{Q}(n) = \text{diag}\{[\mathbf{q}_1^T(n) \ \mathbf{q}_2^T(n)]\}$ is the $2L \times 2L$ diagonal XM tap selection matrix with elements defined by (4.42a) and (4.42b). The subselected and full tap-input matrices each of dimension $K \times 2L$ are then denoted

respectively as

$$\tilde{\mathbf{X}}'_a(n) = [\tilde{\mathbf{x}}'(n) \tilde{\mathbf{x}}'(n-1) \dots \tilde{\mathbf{x}}'(n-K+1)]^T, \quad (4.48)$$

$$\mathbf{X}'_a(n) = [\mathbf{x}'(n) \mathbf{x}'(n-1) \dots \mathbf{x}'(n-K+1)]^T, \quad (4.49)$$

where the subscript a denotes concatenated input vectors for the AP algorithm. Defining $\hat{\mathbf{h}}(n) = [\hat{\mathbf{h}}_1^T(n) \hat{\mathbf{h}}_2^T(n)]^T$ as the $2L \times 1$ two-channel concatenated filter coefficients, the tap update equation for the XMNL-AP algorithm is given as

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + 2\mu \tilde{\mathbf{X}}_a'^T(n) [\mathbf{X}'_a(n) \mathbf{X}'_a'^T(n) + \delta_{\text{AP}} \mathbf{I}_{K \times K}]^{-1} \mathbf{e}(n), \quad (4.50)$$

where $\mathbf{e}(n) = [e(n) e(n-1) \dots e(n-K+1)]^T$ is the $K \times 1$ concatenated *a priori* error with elements computed using (4.6), δ_{AP} is the regularization parameter, μ is the adaptive step-size while $\mathbf{I}_{K \times K}$ is the $K \times K$ identity matrix. It can be seen from (4.50) that for projection order $K = 1$, XMNL-AP is equivalent to XMNL-NLMS.

Note that similar to the single channel MMax-AP algorithm as discussed in Section 2.6.1, XMNL-AP in general cannot be classified as a partial update algorithm since the $2L \times 1$ column vector $\tilde{\mathbf{X}}_a'^T(n) [\mathbf{X}'_a(n) \mathbf{X}'_a'^T(n) + \delta_{\text{AP}} \mathbf{I}_{K \times K}]^{-1} \mathbf{e}(n)$ is a full vector and therefore every element of the adaptive filter $\hat{\mathbf{h}}(n)$ will be updated at each iteration. Special cases may occur if there exist any null rows in the matrix $\tilde{\mathbf{X}}_a'^T(n)$ resulting in a partial adaptation. Such a situation may arise if there are several consecutive small values of $\mathbf{p}(n)$ such that the “inactive” tap-indices in each channel propagate consistently through $\tilde{\mathbf{X}}_a'(n)$ from iteration n to $n - K + 1$. The XMNL-AP algorithm is given in Table 4.3 of Section 4.8.3.

4.5.4 The XMNL-RLS algorithm

Similar to the single channel case, as discussed in Section 2.6.2, direct extension of the XM tap selection approach achieved by sorting the magnitude difference of the Kalman gain $\mathbf{k}(n)$ in the RLS update given by (2.55), will not achieve the desired convergence since the Kalman gain depends on previous values of the time-averaged inverse correlation ma-

trix. The derivation of the XMNL-RLS algorithm is very similar to that of the single channel MMax-RLS algorithm as shown in Section 2.6.2. The two-channel $2L \times 2L$ time-averaged autocorrelation matrix $\Psi(n)$ defined in (4.9) can be expressed in terms of the subselected two-channel tap-input vector $\tilde{\mathbf{x}}'(n) = [\tilde{\mathbf{x}}_1'^T(n) \tilde{\mathbf{x}}_2'^T(n)]^T$ recursively as

$$\begin{aligned}\tilde{\Psi}'(n) &= \tilde{\mathbf{X}}_r'(n)\Lambda(n)\tilde{\mathbf{X}}_r'^T(n) \\ &= \lambda\tilde{\Psi}'(n-1) + \tilde{\mathbf{x}}'(n)\tilde{\mathbf{x}}'^T(n),\end{aligned}\quad (4.51)$$

where $0 << \lambda < 1$ is the forgetting factor and the subscript r in $\tilde{\mathbf{X}}_r'(n)$ denotes concatenated tap-input vectors for the RLS algorithm with

$$\begin{aligned}\tilde{\mathbf{X}}_r'(n) &= [\tilde{\mathbf{x}}'(1) \tilde{\mathbf{x}}'(2) \dots \tilde{\mathbf{x}}'(n)]^T, \\ \Lambda(n) &= \text{diag}\{\lambda^n \lambda^{n-1} \dots \lambda\}.\end{aligned}$$

Similarly, the $2L \times 1$ cross-correlation vector in (4.10) may be expressed recursively as

$$\begin{aligned}\tilde{\Theta}'(n) &= \tilde{\mathbf{X}}_r'(n)\Lambda(n)\mathbf{y}(n) \\ &= \lambda\tilde{\Theta}'(n-1) + \tilde{\mathbf{x}}'(n)y(n),\end{aligned}\quad (4.52)$$

where the concatenated received microphone signal in the receiving room $\mathbf{y}(n)$ is given by

$$\mathbf{y}(n) = [y(1) y(2) \dots y(n)]^T$$

with elements computed using (4.1). Using the matrix inversion lemma and following the approach of [35], the time-averaged input autocorrelation matrix $\tilde{\Psi}'^{-1}(n)$ can be computed using

$$\tilde{\Psi}'^{-1}(n) = \frac{1}{\lambda} \left[\tilde{\Psi}'^{-1}(n-1) - \mathbf{k}(n)\tilde{\mathbf{x}}'^T(n)\tilde{\Psi}'^{-1}(n-1) \right], \quad (4.53)$$

where the $2L \times 1$ modified Kalman gain $\tilde{\mathbf{k}}(n) = [\tilde{\mathbf{k}}_1^T(n) \tilde{\mathbf{k}}_2^T(n)]^T$ is given by

$$\tilde{\mathbf{k}}(n) = \frac{\lambda^{-1}\tilde{\Psi}'^{-1}(n-1)\tilde{\mathbf{x}}'(n)}{1 + \lambda^{-1}\tilde{\mathbf{x}}'^T(n)\tilde{\Psi}'^{-1}(n-1)\tilde{\mathbf{x}}'(n)} \quad (4.54)$$

Defining $e(n)$ as the *a priori* error computed using (4.6), the XMNL-RLS tap update equation is then given by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n)e(n) \quad (4.55)$$

with $\hat{\mathbf{h}}(n) = [\hat{\mathbf{h}}_1^T(n) \hat{\mathbf{h}}_2^T(n)]^T$ being the two-channel filter coefficients. Similar to XMNL-AP, the XMNL-RLS algorithm updates all filter coefficients at each sample iteration since the $2L \times 1$ modified Kalman gain vector $\tilde{\mathbf{k}}(n)$ is a full column vector except in cases where there exist any null rows in $\tilde{\Psi}'^{-1}(n)$. The XMNL-RLS algorithm is summarized in Table 4.4 of Section 4.8.3.

4.5.5 Computational complexity

For comparison purpose, the relative complexity of the algorithms is assessed in terms of the total number of multiplications and comparisons per sample period for each channel. Similar to the MMax-NLMS algorithm, the XMNL-based algorithms employ the SORT-LINE procedure [70] which require at most $2 + 2 \log_2 L$ comparisons. The XMNL-NLMS algorithm requires the same complexity per channel as MMax-NLMS with $M = 0.5L$ and hence, at most $1.5L + 3 + 2 \log_2 L$ operations per sample period per channel is required.

The complexity of AP using the generalized Levinson algorithm is $2LK + 7K^2$ multiplies per sample period [45]. The XMNL-AP algorithm requires an additional $2 + 2 \log_2 L$ sorting operations in each channel for $\tilde{\mathbf{x}}'(n)$. However, due to a reduction in multiplications required when computing $\tilde{\mathbf{X}}_a'^T(n) [\mathbf{X}_a'(n) \mathbf{X}_a'^T(n) + \delta_{\text{AP}} \mathbf{I}_{K \times K}]^{-1}$, the complexity for XMNL-AP is $1.5LK + 7K^2 + 2 + 2 \log_2 L$ operations per sample period per channel.

The number of multiplications required for the RLS algorithm is $4L^2 + 3L + 2$ per adaptive filter where an additional L multiplications are required for the tap updates. Due to the subselection of input vector $\tilde{\mathbf{x}}'(n)$, the number of multiplications required for computing $\tilde{\Psi}'(n)$ for the XMNL-RLS is $1.5L^2 + 1$ while $L(L + 0.5)$ multiplications are required for computing the Kalman gain. Hence the number of operations required for the XMNL-RLS algorithm is at most $2.5L(L + 1) + 3 + 2 \log_2 L$ per sample period per channel.

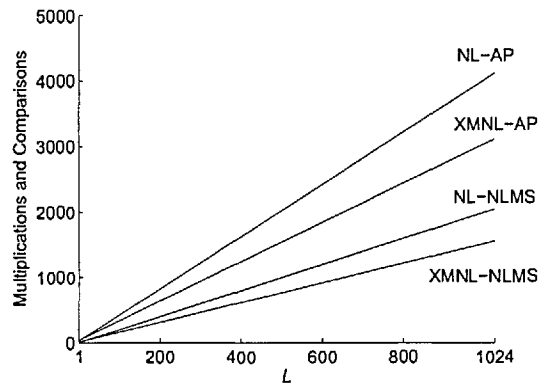


Figure 4.11: Computational complexity of NLMS and AP-based algorithms.

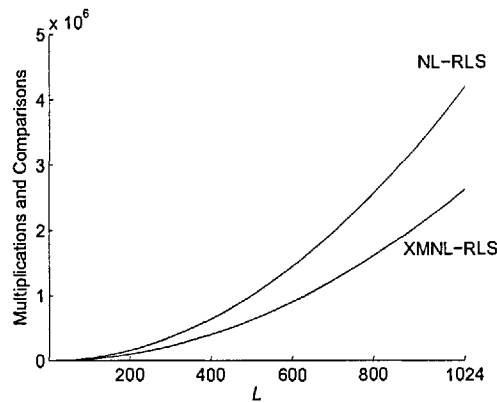


Figure 4.12: Computational complexity of RLS-based algorithms.

Figures 4.11 and 4.12 show the variation of complexity with L for XMNL-NLMS, XMNL-AP and XMNL-RLS algorithms. The projection order for AP-based algorithms is $K = 2$. Although complexity reduction is *not* the main aim of this work, it can be seen that the XM selective-tap techniques nevertheless bring significant computational savings.

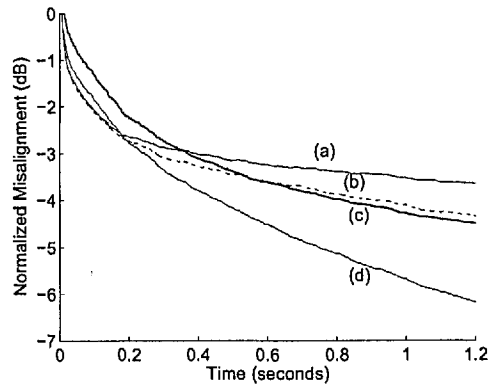


Figure 4.13: Normalized Misalignment for WGN sequence (a) NLMS, (b) NL-NLMS (c) XM-NLMS and (d) XMNL-NLMS [$L_T = 1200$, $L_R = 256$, $L = 256$, $\mu = 0.1$, $\beta = 0.5$, SNR = 25 dB].

4.6 Simulation results

4.6.1 Experimental setup

For all simulations in this section, transmission room impulse responses $\mathbf{g}_1(n)$, $\mathbf{g}_2(n)$ and receiving room impulse responses $\mathbf{h}_1(n)$, $\mathbf{h}_2(n)$ are generated using the method of images [72]. Two microphones are placed 1 m apart in the centre of both the transmission and receiving rooms each of dimension $3 \times 4 \times 5$ m. The source is then positioned 1 m away from each microphone in the transmission room. With reference to Fig. 4.10, tap-input vectors $\mathbf{x}'_1(n)$ and $\mathbf{x}'_2(n)$ are obtained by convolving the source with two impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ and then applying the non-linear (NL) preprocessor defined in (4.46a) and (4.46b). The receiving microphone signal $y(n)$ is obtained using (4.1) and a zero mean WGN sequence $w(n)$ is added such that an arbitrarily chosen SNR of 25 dB is obtained. For clarity, the normalized misalignment of only one channel is plotted in each experiment.

4.6.2 NLMS-based simulations

The performance of XM tap selection and the NL preprocessor in combination with NLMS adaptation is examined. In this experiment, the lengths of the adaptive filters are $L = 256$

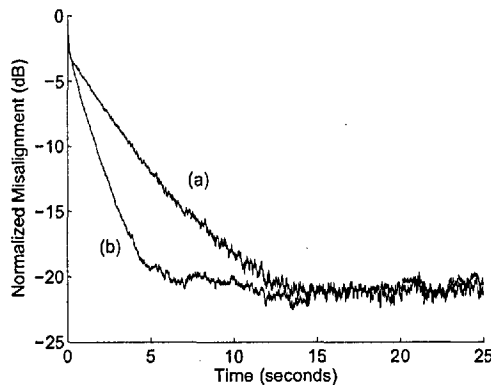


Figure 4.14: Normalized misalignment using WGN input sequence for (a) NL-AP and (b) XMNL-AP [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-AP}} = 0.15$, $\mu_{\text{XMNL-AP}} = 0.1$, $\beta = 0.5$, $K = 2$, $f_s = 8$ kHz, $T_{80} = 100$ ms, SNR = 25 dB].

while the lengths of the transmission and receiving rooms are $L_T = 1600$ and $L_R = 256$ respectively. Figure 4.13 shows the normalized misalignment plot for (a) NLMS, (b) NL-NLMS, (c) XM-NLMS and (d) XMNL-NLMS using a zero mean unit variance WGN source with an arbitrarily chosen step-size of $\mu = 0.1$ for each algorithm. A non-linear distortion factor of $\beta = 0.5$ is used. It can be seen that NLMS has the slowest convergence. The convergence rate of XM-NLMS and NL-NLMS increases significantly due to the XM and NL preprocessors respectively. The XMNL-NLMS algorithm shows even further improvement of approximately 2.5 dB improvement compared to NL-NLMS due to the additional improvement in conditioning caused by XM tap selection. Alternatively, XMNL-NLMS can achieve the same rate of convergence as NL-NLMS but with a lower value of β , hence reducing non-linear distortion [29]. Additional simulation results for XMNL-NLMS can be found in [27].

4.6.3 AP-based simulations

The performance of XMNL-AP is compared with that of the AP algorithm in combination with NL preprocessor (NL-AP) using a zero mean unit variance WGN source sequence. Figure 4.14 shows the normalized misalignment plot for (a) NL-AP and (b) XMNL-AP where a non-linearity factor of $\beta = 0.5$ and a projection order of $K = 2$ are used for both

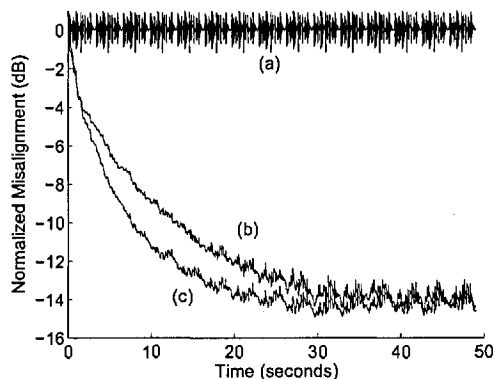


Figure 4.15: Normalized misalignment using (a) speech input sequence for (b) NL-AP and (c) XMNL-AP [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\mu_{\text{NL-AP}} = 0.15$, $\mu_{\text{XMNL-AP}} = 0.1$, $\beta = 0.5$, $K = 2$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

algorithms. The sampling frequency is $f_s = 8$ kHz and the impulse responses used in this experiment are of lengths $L_T = L_R = 800$ with a reverberation time of $T_{60} = 100$ ms. The adaptive filters are each of length $L = 256$ and an uncorrelated zero mean WGN $w(n)$ is added to achieve an SNR of 25 dB. The adaptive step-sizes of the algorithms are chosen to achieve the same steady-state normalized misalignment. For an arbitrary choice of $\mu_{\text{XMNL-AP}} = 0.1$, it is found that for NL-AP, $\mu_{\text{NL-AP}} = 0.15$ gives the same steady-state normalized misalignment as for XMNL-AP. It can be seen from Fig. 4.14 that the XMNL-AP algorithm achieves approximately 5 to 7 dB improvement in normalized misalignment compared to NL-AP during convergence. Alternatively, the NL-AP algorithm requires approximately an additional 10 s before reaching approximately the same steady-state normalized misalignment as XMNL-AP.

The normalized misalignment performance of XMNL-AP is now compared with that of NL-AP using speech signals from a male talker with a sampling frequency of $f_s = 8$ kHz. As before, impulse responses of the transmission and receiving rooms are each of length $L_T = 800$ and $L_R = 800$ respectively with a reverberation time of $T_{60} = 100$ ms and the filters are of length $L = 256$. A projection order of $K = 2$ and a non-linearity constant of $\beta = 0.5$ are used for both NL-AP and XMNL-AP. The step-sizes of NL-AP and XMNL-AP are chosen so that they achieve the same steady-state normalized misalignment. These

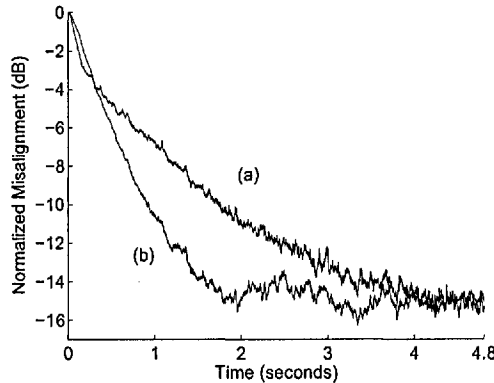


Figure 4.16: Normalized Misalignment with unit variance WGN source for (a) NL-RLS and (b) XMNL-RLS [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\lambda_{\text{XMNL-RLS}} = 0.9987$, $\lambda_{\text{NL-RLS}} = 0.9975$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

step-sizes correspond to $\mu_{\text{NL-AP}} = 0.15$, $\mu_{\text{XMNL-AP}} = 0.1$ for NL-AP and XMNL-AP respectively. An SNR of 25 dB is achieved by adding an uncorrelated zero mean WGN sequence $w(n)$ to the received signal where the SNR is computed using the whole utterance of the speech sequence. Figure 4.15 shows the normalized misalignment for (b) NL-AP and (c) XMNL-AP respectively. As can be seen with this speech input experiment, the XMNL-AP algorithm achieves approximately 2 to 3 dB improvement in normalized misalignment performance compared to NL-AP during convergence. Alternatively, the NL-AP algorithm requires close to an additional 20 s before reaching approximately the same steady-state normalized misalignment as the XMNL-AP algorithm.

4.6.4 RLS-based simulations

The performance of XMNL-RLS is compared with that of the RLS algorithm incorporating the NL preprocessor (NL-RLS) [52] using a zero mean unit variance WGN source sequence. The parameters used in this experiment are $L_T = L_R = 800$, $f_s = 8$ kHz, $T_{60} = 100$ ms, $L = 256$, $M = 128$ and a non-linearity constant of $\beta = 0.5$. As before, an uncorrelated zero mean WGN sequence $w(n)$ is added to the received signal such that an SNR of 25 dB is achieved. A forgetting factor of $\lambda_{\text{XMNL-RLS}} = 1 - [1/(3L)] = 0.9987$ is used [39] for XMNL-RLS while for NL-RLS, $\lambda_{\text{NL-RLS}} = 0.9975$ is used in order for both algorithms

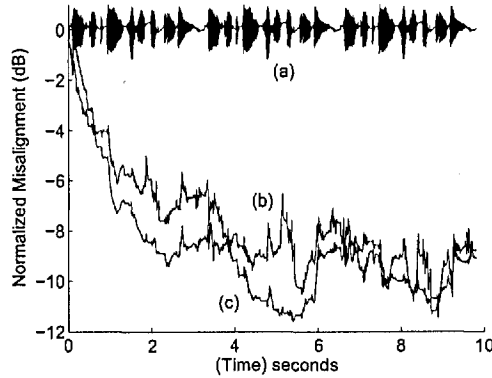


Figure 4.17: (a) Speech signal and normalized misalignment with unit variance WGN source for (b) NL-RLS and (c) XMNL-RLS [$L_T = L_R = 800$, $L = 256$, $M = 128$, $\lambda_{\text{XMNL-RLS}} = 0.99961$, $\lambda_{\text{NL-RLS}} = 0.99957$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

to achieve approximately the same steady-state normalized misalignment. As shown in Fig. 4.16, there is a significant improvement in convergence rate of approximately 4 dB normalized misalignment for the XMNL-RLS algorithm compared to that of NL-RLS during convergence. The NL-RLS algorithm requires close to an additional 2.5 s before reaching approximately the same steady-state normalized misalignment as XMNL-RLS.

Figure 4.17 compares the performances of XMNL-RLS and NL-RLS using a speech source. In this speech experiment, $L_T = L_R = 800$, $T_{60} = 100$ ms, $L = 256$, $M = 128$ and $\beta = 0.5$. The sampling frequency is $f_s = 8$ kHz while the SNR is arbitrarily chosen to be 25 dB with the SNR computed using the whole utterance of the speech sequence. The forgetting factors are $\lambda_{\text{XMNL-RLS}} = 1 - [1/(10L)] = 0.99961$ and $\lambda_{\text{NL-RLS}} = 0.99957$ adjusted experimentally in order for both algorithms to achieve approximately the same steady-state normalized misalignment. It can be seen that due to the additional decorrelating property of XM tap selection, the XMNL-RLS algorithm outperforms NL-RLS by approximately 2 to 4 dB normalized misalignment during convergence.

4.7 Conclusions

In this chapter, a novel tap selection approach has been introduced to improve the conditioning of the input autocorrelation matrix for SAEC application. The “closeness” of

MMax tap selection to the full tap-input vector has been quantified by the M-ratio \mathcal{M} which is then used as an optimization parameter in the development of the proposed XM tap selection technique. It has been shown that the exclusive tap selection criterion reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the input autocorrelation matrix which consequently reduces the misalignment problem. An efficient XM tap selection technique has been developed as an optimization of the MMax criterion (to reduce the degradation in convergence performance due to tap selection) subject to an exclusivity constraint (for reducing interchannel coherence) between the tap selection sets of the two channels. A class of XM-based algorithms have been formulated by applying XM tap selection the NLMS, AP and RLS algorithms for use in combination with the non-linear pre-processing. Simulation results have shown a significant improvement in the range of 2 to 7 dB in convergence compared with algorithms that use the NL-preprocessor alone. Alternatively, the XMNL-based algorithms can be seen to achieve the same convergence performance as NL-based algorithms using a lower non-linear distortion factor β . Although complexity reduction is not the main aim of this work, it has been shown that XM selective-tap updating nevertheless brings significant computational savings.

4.8 Appendix

4.8.1 Proof of non-unique solutions in SAEC

To verify that (4.14) is a valid solution of SAEC which satisfies the relationship $\hat{y}(n) = y(n)$, the z -transformed signals of $\mathbf{g}_1(n)$, $\mathbf{g}_2(n)$ and $s_T(n)$ are defined as $G_1(z)$, $G_2(z)$ and $S_T(z)$ respectively where $s_T(n)$ is the source signal. The z -transform of $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ can then be expressed as

$$X_1(z) = G_1(z)S_T(z) , \quad (4.56)$$

$$X_2(z) = G_2(z)S_T(z) . \quad (4.57)$$

Thus it can be seen that $X_1(z)G_2(z) = X_2(z)G_1(z)$ hence giving

$$\mathbf{x}_1^T(n)\mathbf{g}_2(n) = \mathbf{x}_2^T(n)\mathbf{g}_1(n). \quad (4.58)$$

The output of the adaptive filters $\hat{y}(n)$ may be then be expressed, using (4.14) and (4.58), as

$$\begin{aligned} \hat{y}(n) &= \hat{\mathbf{h}}_1^T(n)\mathbf{x}_1(n) + \hat{\mathbf{h}}_2^T(n)\mathbf{x}_2(n) \\ &= [\mathbf{h}_1(n) + \varphi(n)\mathbf{g}_2(n)]^T \mathbf{x}_1(n) + [\mathbf{h}_2(n) - \varphi(n)\mathbf{g}_1(n)]^T \mathbf{x}_2(n) \\ &= y(n) \end{aligned} \quad (4.59)$$

hence giving the *a posteriori* error $e_p(n) = 0$ for any scalar $\varphi(n)$.

4.8.2 Verification of exclusive tap selection set which maximizes \mathcal{M}

In the following, the dependency of sample iteration n has been removed for clarity in notation. For illustration purpose, it is also assumed that $L \bmod 2 = 0$ where *mod* is the modulo operator. To verify that the exclusive tap selection set given by (4.41) maximizes \mathcal{M} jointly for both channels at each sample iteration, it is necessary to consider whether the absolute sum of the XM tap selection given by $\sum_{i=0}^{M-1} |\check{x}_{1,i}| + \sum_{i=M}^{L-1} |\check{x}_{2,i}|$ is greater than the absolute sum obtained from all ${}^L C_M - 1$ other exclusive combinations of tap-inputs where $M = 0.5L$. This can be achieved by first testing whether the condition

$$\sum_{i=0}^{M-1} |\check{x}_{1,i}| + \sum_{i=M}^{L-1} |\check{x}_{2,i}| > \sum_{i=M}^{L-1} |\check{x}_{1,i}| + \sum_{i=0}^{M-1} |\check{x}_{2,i}| \quad (4.60)$$

holds, where the left-hand-side terms corresponds to the XM tap selection. Using the definition of \check{p}_i given in (4.39), (4.60) can be simplified, giving

$$\begin{aligned} \sum_{i=0}^{M-1} \left[|\check{x}_{1,i}| - |\check{x}_{2,i}| \right] &> \sum_{i=M}^{L-1} \left[|\check{x}_{1,i}| - |\check{x}_{2,i}| \right] \\ \sum_{i=0}^{M-1} \check{p}_i &> \sum_{i=M}^{L-1} \check{p}_i \end{aligned} \quad (4.61)$$

which is valid from the definition of $\check{\mathfrak{p}}$. The ${}^L C_{M-2}$ other possible cases can be considered using the above approach. Suppose for example, tap-indices in the set $\{\check{x}_{1,2i}\}$ and $\{\check{x}_{2,2i+1}\}$ are now selected, where $i = 0, 1, \dots, M-1$ for which the condition

$$\sum_{i=0}^{M-1} |\check{x}_{1,i}| + \sum_{i=M}^{L-1} |\check{x}_{2,i}| > \sum_{i=0}^{M-1} |\check{x}_{1,2i}| + \sum_{i=0}^{M-1} |\check{x}_{2,2i+1}| \quad (4.62)$$

must now be verified. Rewriting (4.62),

$$\sum_{i=0}^{M-1} |\check{x}_{1,i}| - \sum_{i=0}^{M-1} |\check{x}_{2,2i+1}| > \sum_{i=0}^{M-1} |\check{x}_{1,2i}| - \sum_{i=M}^{L-1} |\check{x}_{2,i}| \quad (4.63)$$

is obtained, from which each term can be decomposed as

$$\sum_{i=0}^{M-1} |\check{x}_{1,i}| = \sum_{i=0}^{M/2-1} |\check{x}_{1,2i}| + \sum_{i=0}^{M/2-1} |\check{x}_{1,2i+1}|, \quad (4.64a)$$

$$\sum_{i=0}^{M-1} |\check{x}_{2,2i+1}| = \sum_{i=0}^{M/2-1} |\check{x}_{2,2i+1}| + \sum_{i=M/2}^{M-1} |\check{x}_{2,2i+1}|, \quad (4.64b)$$

$$\sum_{i=0}^{M-1} |\check{x}_{1,2i}| = \sum_{i=0}^{M/2-1} |\check{x}_{1,2i}| + \sum_{i=M/2}^{M-1} |\check{x}_{1,2i}|, \quad (4.64c)$$

$$\sum_{i=M}^{L-1} |\check{x}_{2,i}| = \sum_{i=M/2}^{M-1} |\check{x}_{2,2i}| + \sum_{i=M/2}^{M-1} |\check{x}_{2,2i+1}|. \quad (4.64d)$$

Using (4.64a)-(4.64d), (4.63) can be simplified as

$$\begin{aligned} \sum_{i=0}^{M/2-1} \left[|\check{x}_{1,2i+1}| - |\check{x}_{2,2i+1}| \right] &> \sum_{i=M/2}^{M-1} \left[|\check{x}_{1,2i}| - |\check{x}_{2,2i}| \right] \\ \sum_{i=0}^{M/2-1} \check{\rho}_{2i+1} &> \sum_{i=M/2}^{M-1} \check{\rho}_{2i} \\ \sum_{i=0}^{M/2-1} \check{\rho}_{2i+1} &> \sum_{i=0}^{M/2-1} \check{\rho}_{2i+M}. \end{aligned} \quad (4.65)$$

With $M \geq 1$ and from the definition of $\check{\mathfrak{p}}$, (4.65) and consequently (4.62) is valid. Similar analysis can then be used to verify the remaining cases.

4.8.3 XMNL-based algorithms for SAEC

Table 4.1: The XMNL tap selection

M	$= 0.5L$
$\mathbf{x}'_1(n)$	$= \mathbf{x}_1(n) + 0.5\beta[\mathbf{x}_1(n) + \mathbf{x}_1(n)]$
$\mathbf{x}'_2(n)$	$= \mathbf{x}_2(n) + 0.5\beta[\mathbf{x}_2(n) - \mathbf{x}_2(n)]$
$\mathbf{x}'(n)$	$= [\mathbf{x}'_1{}^T(n) \ \mathbf{x}'_2{}^T(n)]^T$
$\hat{\mathbf{h}}(n)$	$= [\hat{\mathbf{h}}_1{}^T(n) \ \hat{\mathbf{h}}_2{}^T(n)]^T$
$\mathbf{p}(n)$	$= \mathbf{x}'_1(n) - \mathbf{x}'_2(n) $
$\mathbf{Q}(n)$	$= \text{diag}\{[\mathbf{q}_1{}^T(n) \ \mathbf{q}_2{}^T(n)]\}$
$q_{1,u}(n)$	$= \begin{cases} 1, & p_u(n) \in \{M \text{ maxima of } \mathbf{p}(n)\} \\ 0, & \text{otherwise} \end{cases}$
$q_{2,v}(n)$	$= \begin{cases} 1, & p_v(n) \in \{M \text{ minima of } \mathbf{p}(n)\} \\ 0, & \text{otherwise} \end{cases}$

Table 4.2: The XMNL-NLMS algorithm

$\hat{y}(n)$	$= \hat{\mathbf{h}}^T(n-1)\mathbf{x}'(n)$
$e(n)$	$= y(n) - \hat{y}(n)$
$\hat{\mathbf{h}}(n)$	$= \hat{\mathbf{h}}(n-1) + \mathbf{Q}(n) \frac{2\mu\mathbf{x}'(n)e(n)}{\ \mathbf{x}'(n)\ _2^2 + \delta_{\text{NLMS}}}$

Table 4.3: The XMNL-AP algorithm

$$\begin{aligned}
\mathbf{X}'_a(n) &= [\mathbf{x}'(n) \mathbf{x}'(n-1) \dots \mathbf{x}'(n-K+1)]^T \\
\tilde{\mathbf{x}}'(n) &= \mathbf{Q}(n)\mathbf{x}'(n) \\
\tilde{\mathbf{X}}'_a(n) &= [\tilde{\mathbf{x}}'(n) \tilde{\mathbf{x}}'(n-1) \dots \tilde{\mathbf{x}}'(n-K+1)]^T \\
\mathbf{y}(n) &= [y(n) y(n-1) \dots y(n-K+1)]^T \\
\hat{\mathbf{y}}(n) &= \mathbf{X}'_a(n)\hat{\mathbf{h}}(n-1) \\
\mathbf{e}(n) &= \mathbf{y}(n) - \hat{\mathbf{y}}(n) \\
\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + 2\mu\tilde{\mathbf{X}}'^T_a(n)[\mathbf{X}'_a(n)\mathbf{X}'_a{}^T(n) + \delta_{\text{AP}}\mathbf{I}_{K \times K}]^{-1}\mathbf{e}(n)
\end{aligned}$$

Table 4.4: The XMNL-RLS algorithm

$$\begin{aligned}
\tilde{\Psi}'^{-1}(0) &= \delta_{\text{RLS}}^{-1}\mathbf{I}_{L \times L} \\
\tilde{\mathbf{k}}(n) &= [\tilde{\mathbf{k}}_1^T(n) \tilde{\mathbf{k}}_2^T(n)]^T \\
\tilde{\mathbf{x}}'(n) &= \mathbf{Q}(n)\mathbf{x}'(n) \\
\tilde{\mathbf{k}}(n) &= \frac{\tilde{\Psi}'^{-1}(n-1)\tilde{\mathbf{x}}'(n)}{\lambda + \tilde{\mathbf{x}}'^T(n)\tilde{\Psi}'^{-1}(n-1)\tilde{\mathbf{x}}'(n)} \\
\hat{\mathbf{y}}(n) &= \hat{\mathbf{h}}^T(n-1)\mathbf{x}'(n) \\
\mathbf{e}(n) &= \mathbf{y}(n) - \hat{\mathbf{y}}(n) \\
\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + \tilde{\mathbf{k}}(n)\mathbf{e}(n) \\
\tilde{\Psi}'^{-1}(n) &= \frac{1}{\lambda}[\tilde{\Psi}'^{-1}(n-1) - \tilde{\mathbf{k}}(n)\tilde{\mathbf{x}}'^T(n)\tilde{\Psi}'^{-1}(n-1)]
\end{aligned}$$

Chapter 5

Frequency-Domain Algorithms with Applications to Stereophonic Acoustic Echo Cancellation

*For the things we have to learn before we
can do them, we learn by doing them.*

Aristotle (384-322 BC)

5.1 Introduction

FREQUENCY-DOMAIN adaptive filtering has been increasingly popular in recent years and was introduced as a form of improving the efficiency of time-domain algorithms. As opposed to time-domain algorithms, such as LMS, NLMS, AP and RLS as discussed in previous chapters, where computation is performed sample-by-sample, frequency-domain algorithms generally inherit two properties (i) incorporating block updating strategies and (ii) employing the fast Fourier transform (FFT). A direct consequence of block processing is the reduction in computational complexity since the filter output and tap updates are computed only after a block of data has been accumulated. In addition, the use of the FFT for computing the discrete Fourier transform (DFT) so as to perform linear convolution and gradient estimation further increases the efficiency of such

algorithms.

The concept of frequency-domain adaptive filtering was first introduced in [105] where signals $x(n)$ and $y(n)$, as depicted in Fig. 2.1, are accumulated in buffer memories to form data blocks which in turn are transformed to the frequency-domain using FFTs. Defining L as the length of the adaptive filter, it has been found that the resultant frequency-domain LMS algorithm achieves a reduction in computation such that the ratio between complex multiplies in frequency-domain LMS to real multiplies in conventional LMS reduces significantly for large L . For $L = 16$, it has been shown [105] that this ratio is approximately 0.25. It has been noted however that the frequency-domain LMS algorithm converges to a sub-optimal Wiener solution due to the effects of circular convolution [106] [107]. To address this, the fast-LMS (FLMS) algorithm was proposed [22] where the overlap-save method of implementing linear convolution using FFT blocks is employed. Although five $2L$ -point FFT blocks are required, the FLMS algorithm achieves reduction in computation by approximately a factor of 4 in terms of complex multiplies compared to real multiplies for LMS when $L = 256$. The block-LMS (BLMS) algorithm [108] was derived independently using the block mean-square error (BMSE) and as before, implemented using FFTs. The unconstrained-FLMS (UFLMS) algorithm was proposed in [109] using only three FFT blocks and converges to the Wiener solution under the condition that the length of unknown system is less than or equal to L .

Although substantial computational savings can be achieved, one of the main drawbacks of frequency-domain approaches is the inherent delay introduced between the input and output. This delay corresponds to the length of the adaptive filter L since filter outputs are computed frame-by-frame after every L samples. On the other hand, as explained in Section 2.2.2, in order to achieve sufficient misalignment performance, L must be large enough and consequently research in recent years have been focusing on reducing the delay of such frequency-domain algorithms. To mitigate the problem of delay, the multi-delay filter (MDF) structure was proposed [23] which partitions the adaptive filter into \mathcal{K} blocks each of length N such that $L = \mathcal{K}N$. A general approach based on the weighted-overlap-and-add (WOLA) method was proposed in [110]. The resulting general-

ized multi-delay filter (GMDF α) introduce an additional degree of freedom by employing an arbitrary overlapping factor between successive input frames controlled by $\alpha \geq 1$ such that for $\alpha = 1$, a 50% input overlapping factor is achieved. It has been noted that even though, from complexity point of view, the optimal choice is $L = N$, using $N < L$ is still more efficient than time-domain algorithms [11].

It is well-known that for a single channel system identification case, the performance of an adaptive algorithm is affected by the conditioning of the input autocorrelation matrix \mathbf{R}_{xx} . Specifically, as the condition number increases, the rate of convergence is reduced and poor misalignment is exhibited. A recursive estimation of the condition number in the single channel RLS algorithm has been proposed in [111] [112] where it has been shown how misalignment of the RLS algorithm is degraded by the increase in condition number. For the stereo case, as discussed in Section 4.3.2 and reported in [11] [90] [91] [92] [113] [114], although it has been noted that the conditioning of \mathbf{R}_{xx} is degraded by the high interchannel coherence between the two input signals $x_1(n)$ and $x_2(n)$ as depicted in Fig. 4.1, no explicit relationship between the two has yet been established.

The main contribution of this chapter is two-fold: (i) to establish the relationship between interchannel coherence and condition number of \mathbf{R}_{xx} for stereophonic acoustic echo cancellation (SAEC) and (ii) to extend the exclusive-maximum (XM) tap selection, as developed in Chapter 4, to the frequency-domain adaptive filtering for efficient implementation. The relationship between interchannel coherence and conditioning of the input autocorrelation matrix allows one to gain further insight into how interchannel coherence degrades the misalignment performance of SAEC algorithms through the ill-conditioning of the input autocorrelation matrix. Using this relationship, one can determine the level of ill-conditioning and design parameters so as to improve the conditioning of the autocorrelation matrix hence giving good misalignment performance such as shown in [113]. As has been shown in Chapter 4, the XM selective-tap algorithms can achieve good convergence performance. Tap selection for frequency-domain adaptive algorithms can be achieved by first considering subselection either in the time-domain or frequency-domain. In this chapter, these cases will be considered and their effect on the convergence rate will be

examined in the context of the M-ratio measure \mathcal{M} .

This chapter is organized as follows: In Section 5.2, data sectioning matrices that are commonly used in this chapter are defined. The single channel FLMS and MDF algorithms are reviewed in Section 5.3 which, in addition, allow frequency-domain variables associated with this chapter to be defined. Following closely the approach presented in [11], a generalized derivation of these algorithms using a frequency-domain cost function is reviewed. In Section 5.4, the link between interchannel coherence and the conditioning of \mathbf{R}_{xx} is established for the stereophonic case. The XM tap selection is then extended, in Section 5.5, to the FLMS algorithm employing both the 50% and an arbitrary overlapping factor between successive tap-input vectors for efficient implementation. Simulation results are provided in Section 5.6 to verify theoretical analysis and to compare the performances of the proposed algorithms.

5.2 Definition of data-sectioning and commonly used matrices

For reasons of clarity, commonly used matrices in this chapter are defined in this section. For consistency, notations found in [11] are adopted for these matrices. The $N \times N$ identity matrix is denoted as $\mathbf{I}_{N \times N}$ while $\mathbf{0}_{N \times N}$ is a null matrix of the same dimension. The $2L \times 2L$ Fourier matrix [106] is defined as

$$\mathbf{F}_{2L \times 2L} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-i2\pi/2L} & e^{-i4\pi/2L} & \dots & e^{-i2\pi(2L-1)/2L} \\ 1 & e^{-i4\pi/2L} & e^{-i8\pi/2L} & \dots & e^{-i4\pi(2L-1)/2L} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & e^{-i2\pi(2L-1)/2L} & e^{-i4\pi(2L-1)/2L} & \dots & e^{-i2\pi(2L-1)^2/2L} \end{bmatrix}_{2L \times 2L}, \quad (5.1)$$

where $i = \sqrt{-1}$ in this chapter and for practical implementation, $\mathbf{F}_{2L \times 2L}^{-1} = [1/(2L)]\mathbf{F}_{2L \times 2L}^*$ where * denotes complex conjugation [106]. The following matrices are

now defined

$$\mathbf{W}_{2N \times N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} \\ \mathbf{I}_{N \times N} \end{bmatrix}_{2N \times N}, \quad (5.2a)$$

$$\mathbf{W}_{N \times 2N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \end{bmatrix}_{N \times 2N}, \quad (5.2b)$$

$$\mathbf{W}_{2N \times N}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix}_{2N \times N}, \quad (5.2c)$$

$$\mathbf{W}_{L \times 2L}^{10} = \begin{bmatrix} \mathbf{I}_{L \times L} & \mathbf{0}_{L \times L} \end{bmatrix}_{L \times 2L}, \quad (5.2d)$$

$$\mathbf{W}_{2L \times 2L}^{10} = \begin{bmatrix} \mathbf{I}_{L \times L} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{0}_{L \times L} \end{bmatrix}_{2L \times 2L}, \quad (5.2e)$$

$$\mathbf{W}_{2L \times 2L}^{01} = \begin{bmatrix} \mathbf{0}_{L \times L} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{I}_{L \times L} \end{bmatrix}_{2L \times 2L}, \quad (5.2f)$$

$$\mathbf{G}_{2L \times 2L}^{10} = \mathbf{F}_{2L \times 2L} \mathbf{W}_{2L \times 2L}^{10} \mathbf{F}_{2L \times 2L}^{-1}, \quad (5.2g)$$

$$\mathbf{G}_{2L \times 2L}^{01} = \mathbf{F}_{2L \times 2L} \mathbf{W}_{2L \times 2L}^{01} \mathbf{F}_{2L \times 2L}^{-1}, \quad (5.2h)$$

$$\mathbf{G}_{N \times 2N}^{01} = \mathbf{F}_{N \times N} \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1}, \quad (5.2i)$$

$$\mathbf{G}_{2N \times N}^{10} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10} \mathbf{F}_{N \times N}^{-1}, \quad (5.2j)$$

$$\mathbf{G}_{2N \times 2N}^{10} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{10} \mathbf{F}_{2N \times 2N}^{-1}, \quad (5.2k)$$

$$\mathbf{G}_{2N \times 2N}^{01} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1}. \quad (5.2l)$$

It should be noted that the variables L and N denote the size of each matrix and may be substituted interchangeably.

5.3 Single channel frequency-domain adaptive algorithms: A review

This section provides a review of existing techniques for single channel frequency-domain adaptive filters. Whilst this section contain no new material, it is introduced here to establish the analytical framework that will be used in later sections and to define the notation

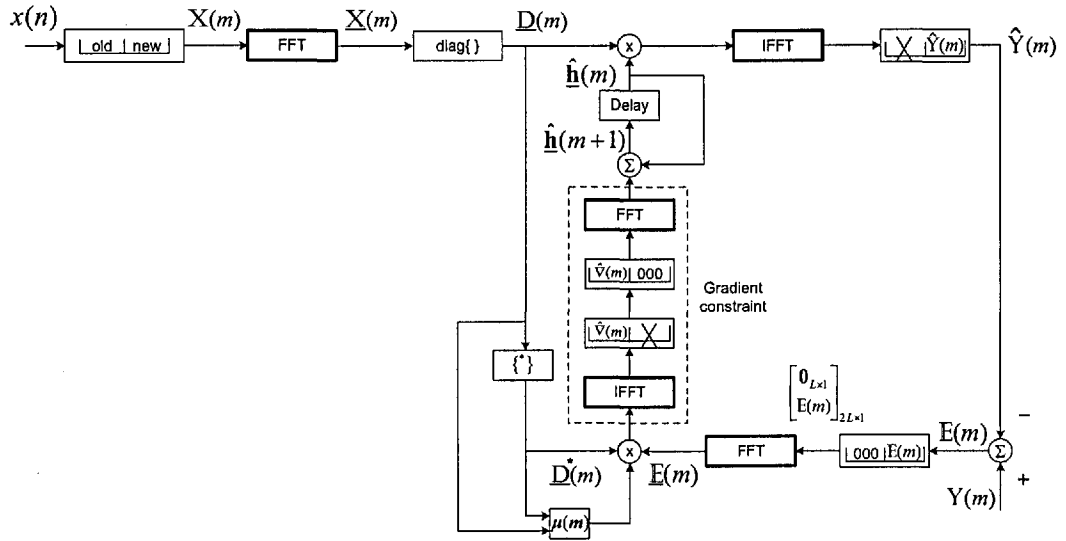


Figure 5.1: Schematic of the FLMS algorithm (after [115]).

employed. Readers already familiar with this topic can move safely on to Section 5.4 using Section 5.3 purely as a reference for notational purposes.

5.3.1 The fast-LMS adaptive algorithm

The fast-LMS (FLMS) algorithm, as depicted in Fig. 5.1, was proposed in [22] employing a block-based updating strategy. The main difference between FLMS and frequency-domain LMS [105] is that the former introduces two additional FFT blocks for gradient computation and employs the overlap-save method [107] for linear convolution. In contrast to the sample-by-sample algorithms such as LMS, the FLMS algorithm adapts its filter coefficients by first concatenating input signal $x(n)$ into frames and employ a 50% overlapping factor between successive frames as shown in Fig. 5.2. These frames are transformed into their discrete Fourier transformed (DFT) sequences using the fast Fourier transform (FFT) algorithm for efficient implementation [116] [117]. Defining m as the frame-index for $m = 0, 1, \dots$, the m^{th} input frame, of dimension $2L \times 1$, is given by

$$\mathbb{X}(m) = [x(mL-L) \ x(mL-L+1) \ \dots \ x(mL-1) \ x(mL) \ x(mL+1) \ \dots \ x(mL+L-1)]^T \quad (5.3)$$

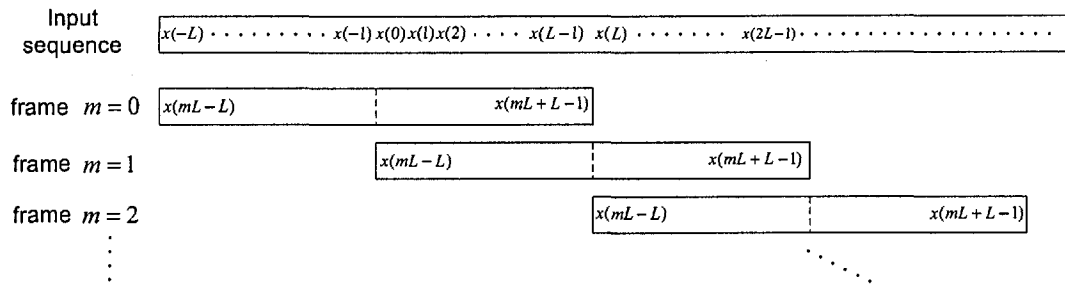


Figure 5.2: Input sequence partitioning for the FLMS algorithm with 50% overlap between successive frames.

while the $L \times 1$ estimated impulse response is given by

$$\hat{\mathbf{h}}(m) = [\hat{h}_0(m) \hat{h}_1(m) \dots \hat{h}_{L-1}(m)]^T. \quad (5.4)$$

The frequency-domain input sequence can be expressed as

$$\begin{aligned} \underline{\mathbf{X}}(m) &= \mathbf{F}_{2L \times 2L} \mathbf{X}(m) \\ &= [\underline{x}_0(m) \underline{x}_1(m) \dots \underline{x}_{2L-1}(m)]^T, \end{aligned} \quad (5.5)$$

where $\underline{x}_l(m)$ is the l^{th} frequency-bin of the input signal for $l = 0, 1, \dots, 2L - 1$. In this chapter, for reasons of clarity and unless explicitly stated, all frequency-domain variables are denoted with an underscore. The $L \times 1$ received microphone signal is given by

$$\mathbf{Y}(m) = [y(mL) y(mL+1) \dots y(mL+L-1)]^T. \quad (5.6)$$

Since time-domain convolution can be expressed using multiplication in the frequency-domain, a $2L \times 2L$ diagonal matrix

$$\begin{aligned} \underline{\mathbf{D}}(m) &= \text{diag}\{\underline{\mathbf{X}}(m)\} \\ &= \begin{bmatrix} \underline{x}_0(m) & 0 & \dots & 0 \\ 0 & \underline{x}_1(m) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \underline{x}_{2L-1}(m) \end{bmatrix}_{2L \times 2L} \end{aligned} \quad (5.7)$$

can be defined containing the frequency content of the input sequence and

$$\hat{\underline{\mathbf{h}}}(m) = \mathbf{F}_{2L \times 2L} \begin{bmatrix} \hat{\mathbf{h}}(m) \\ \mathbf{0}_{L \times 1} \end{bmatrix}_{2L \times 1} \quad (5.8)$$

is the frequency-domain estimated response where $\mathbf{0}_{L \times 1}$ is the $L \times 1$ null vector. The element-by-element multiplication between $\underline{\mathbf{X}}(m)$ and $\hat{\underline{\mathbf{h}}}(m)$ can then be expressed as $\underline{\mathbf{D}}(m)\hat{\underline{\mathbf{h}}}(m)$. Noting that only the last L terms of $\underline{\mathbf{D}}(m)\hat{\underline{\mathbf{h}}}(m)$ correspond to linear convolution and using $\mathbf{G}_{2L \times 2L}^{01}$ as defined by (5.2h), the frequency-domain output of the adaptive filter can be expressed as

$$\begin{aligned} \hat{\underline{\mathbf{Y}}}(m) &= \mathbf{F}_{2L \times 2L} \begin{bmatrix} \mathbf{0}_{L \times 1} \\ \hat{\underline{\mathbf{Y}}}(m) \end{bmatrix}_{2L \times 1} \\ &= \mathbf{G}_{2L \times 2L}^{01} \underline{\mathbf{D}}(m) \hat{\underline{\mathbf{h}}}(m-1), \end{aligned} \quad (5.9)$$

where

$$\hat{\underline{\mathbf{Y}}}(m) = [\hat{y}(mL) \hat{y}(mL+1) \dots \hat{y}(mL+L-1)]^T. \quad (5.10)$$

Consequently, defining the time-domain *a priori* block error as

$$\begin{aligned} \underline{\mathbf{E}}(m) &= [e(mL) e(mL+1) \dots e(mL+L-1)]^T \\ &= \underline{\mathbf{Y}}(m) - \hat{\underline{\mathbf{Y}}}(m), \end{aligned} \quad (5.11)$$

the $2L \times 1$ frequency-domain *a priori* error is then

$$\begin{aligned} \underline{\mathbf{E}}(m) &= \mathbf{F}_{2L \times 2L} \begin{bmatrix} \mathbf{0}_{L \times 1} \\ \underline{\mathbf{E}}(m) \end{bmatrix}_{2L \times 1} \\ &= \underline{\mathbf{Y}}(m) - \hat{\underline{\mathbf{Y}}}(m) \\ &= \underline{\mathbf{Y}}(m) - \mathbf{G}_{2L \times 2L}^{01} \underline{\mathbf{D}}(m) \hat{\underline{\mathbf{h}}}(m-1), \end{aligned} \quad (5.12)$$

where, using (5.6),

$$\underline{\mathbf{Y}}(m) = \mathbf{F}_{2L \times 2L} \begin{bmatrix} \mathbf{0}_{L \times 1} \\ \mathbf{Y}(m) \end{bmatrix}_{2L \times 1} \quad (5.13)$$

It has been known that the convergence rate of gradient descent algorithms is a reducing function of the eigenvalue spread of the input autocorrelation matrix [35]. For a white Gaussian noise (WGN) sequence, these eigenvalues correspond approximately to the energy of the signal spectrum at equally spaced frequency points around the unit-circle [118] [119]. To compensate for this energy variation, the FLMS algorithm weighs each frequency-bin such that the effective step-size for each element in the gradient vector is inversely proportional to the energy of the input signal at that frequency-bin. As a result, a more uniform convergence can be achieved across different frequency-bins. Defining $*$ as the conjugate operator, this energy can be estimated recursively using a $2L \times 2L$ matrix [120] [121]

$$\begin{aligned} \underline{\mathcal{P}}_{\text{FLMS}}(m) &= \lambda \underline{\mathcal{P}}_{\text{FLMS}}(m-1) + (1-\lambda) \underline{\mathbb{D}}^*(m) \underline{\mathbb{D}}(m) \\ &= \text{diag} \left\{ \underline{\mathcal{P}}_0(m) \underline{\mathcal{P}}_1(m) \dots \underline{\mathcal{P}}_{2L-1}(m) \right\}, \end{aligned} \quad (5.14)$$

where $\underline{\mathcal{P}}_l(m)$ is the l^{th} frequency-bin energy content of the input signal and $0 << \lambda < 1$ is the forgetting factor while the diagonal matrix $\underline{\mathbb{D}}(m)$ is defined in (5.7). The frequency-dependent step-size is then given by

$$\mu(m) = 2\mu(1-\lambda) \left[\underline{\mathcal{P}}_{\text{FLMS}}(m) + \delta_{\text{FLMS}} \mathbf{I}_{2L \times 2L} \right]^{-1}, \quad (5.15)$$

where $\mathbf{I}_{2L \times 2L}$ is the $2L \times 2L$ identity matrix, $0 < \mu \leq 1$ is the step-size and δ_{FLMS} is the regularization parameter [11].

Similar to time-domain adaptive filtering as discussed in Section 2.4, the FLMS, being a stochastic gradient descent algorithm, employs a gradient estimate given by the correlation between the *a priori* error and the input sequence. As opposed to the output of the adaptive filter where the *last* L terms of $\underline{\mathbb{D}}(m) \hat{\underline{\mathbf{h}}}(m)$ corresponds to linear convolution,

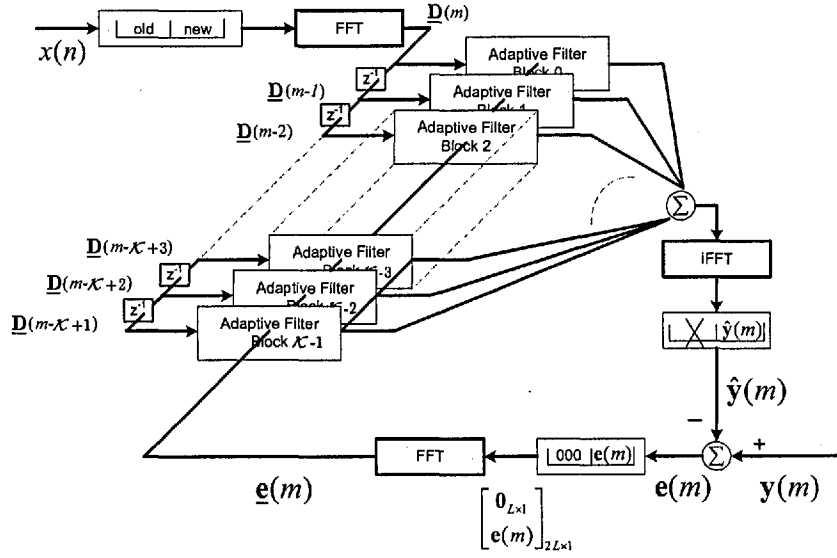


Figure 5.3: Schematic of the MDF structure (after [23]).

only the *first* L terms in $\mathbf{F}_{2L \times 2L}^{-1} \mathbb{D}^*(m) \mathbb{E}(m)$ correspond to the gradient estimate given by

$$\hat{\mathbf{V}}(m) = \mathbf{W}_{L \times 2L}^{10} \mathbf{F}_{2L \times 2L}^{-1} \mathbb{D}^*(m) \mathbb{E}(m), \quad (5.16)$$

where the windowing matrix $\mathbf{W}_{L \times 2L}^{10}$, defined in (5.2d), selects the first L elements of $\mathbf{F}_{2L \times 2L}^{-1} \mathbb{D}^*(m) \mathbb{E}(m)$. The frequency-domain update equation of the FLMS algorithm can then be expressed by

$$\begin{aligned} \hat{\mathbf{h}}(m) &= \hat{\mathbf{h}}(m-1) + \mathbf{G}_{2L \times 2L}^{10} \mu(m) \mathbb{D}^*(m) \mathbb{E}(m) \\ &= \hat{\mathbf{h}}(m-1) + 2\mu(1-\lambda) \mathbf{G}_{2L \times 2L}^{10} \mathbb{D}^*(m) [\mathcal{P}_{\text{FLMS}}(m) + \delta_{\text{FLMS}} \mathbf{I}_{2L \times 2L}]^{-1} \mathbb{E}(m), \end{aligned} \quad (5.17)$$

where $\mathbf{G}_{2L \times 2L}^{10}$ and the frequency-domain *a priori* error $\mathbb{E}(m)$ are defined in (5.2g) and (5.12) respectively. The FLMS algorithm is summarized in Table 5.2 of Section 5.8.7.

5.3.2 The multi-delay filtering (MDF) structure

The multi-delay filtering (MDF) structure [23], as shown in Fig. 5.3, was proposed to mitigate the problem of delay inherent in FLMS since, as can be seen from Section 5.3.1, the FLMS algorithm computes the output $\hat{\mathbf{Y}}(m)$ for every L input samples. The MDF

structure addresses the problem of delay by partitioning the adaptive filter of length L into \mathcal{K} blocks each having length N such that $L = \mathcal{K}N$. Consequently, the delay of the MDF structure is reduced by a factor of L/N compared to the FLMS algorithm. As a consequence of a smaller block size for $N < L$, filter coefficients are updated more frequently (once every N samples compared to L for FLMS) hence achieving faster convergence. For each block, the filter coefficients are then updated similarly to the FLMS algorithm employing the FFT algorithm for linear convolution and gradient estimation as discussed in Section 5.3.1. The MDF structure has also been proposed for sparse system identification as shown in [33].

The single channel MDF structure can be described by first defining, for the m^{th} frame, $m = 0, 1, \dots$, the input block sequence

$$\mathbf{x}(mN) = [x(mN) \ x(mN - 1) \ \dots \ x(mN - L + 1)]^T. \quad (5.18)$$

Concatenating offset versions of this input sequence N times, the matrix

$$\begin{aligned} \mathbf{X}(m) &= [\mathbf{x}(mN) \ \mathbf{x}(mN + 1) \ \dots \ \mathbf{x}(mN + N - 1)]_{L \times N} \\ &= \begin{bmatrix} x(mN) & x(mN + 1) & \dots & x(mN + N - 1) \\ x(mN - 1) & x(mN) & \dots & x(mN + N - 2) \\ \vdots & \vdots & \vdots & \vdots \\ x(mN - L + 1) & x(mN - L + 2) & \dots & x(mN + N - L) \end{bmatrix}_{L \times N} \end{aligned} \quad (5.19)$$

is obtained, from which using the definition of the estimated impulse response $\hat{\mathbf{h}}(m)$ defined by (5.4), the filter output can then be expressed as convolution between the input sequence and the filter coefficients given by the $N \times 1$ vector

$$\begin{aligned} \hat{\mathbf{y}}(m) &= \mathbf{X}^T(m) \hat{\mathbf{h}}(m - 1) \\ &= \begin{bmatrix} x(mN) & x(mN - 1) & \dots & x(mN - L + 1) \\ x(mN + 1) & x(mN) & \dots & x(mN - L + 2) \\ \vdots & \vdots & \vdots & \vdots \\ x(mN + N - 1) & x(mN + N - 2) & \dots & x(mN + N - L) \end{bmatrix}_{N \times L} \begin{bmatrix} \hat{h}_0(m - 1) \\ \hat{h}_1(m - 1) \\ \vdots \\ \hat{h}_{L-1}(m - 1) \end{bmatrix}_{L \times 1} \\ &= [\hat{\mathbf{y}}(mN) \ \hat{\mathbf{y}}(mN + 1) \ \dots \ \hat{\mathbf{y}}(mN + N - 1)]^T. \end{aligned} \quad (5.20)$$

Defining

$$\mathbf{y}(m) = [y(mN) \ y(mN + 1) \ \dots \ y(mN + N - 1)]^T \quad (5.21)$$

as the $N \times 1$ received microphone signal, the *a priori* block error can be expressed as

$$\begin{aligned} \mathbf{e}(m) &= \mathbf{y}(m) - \hat{\mathbf{y}}(m) \\ &= [e(mN) \ e(mN + 1) \ \dots \ e(mN + N - 1)]^T. \end{aligned} \quad (5.22)$$

Note that for a single block $\mathcal{K} = 1$, the block size is equivalent to the length of the adaptive filter, i.e., $N = L$ and consequently, (5.22) is equivalent to (5.11). The rectangular matrix $\mathbf{X}^T(m)$ can be decomposed [122] into sub-matrices each of size $N \times N$ while the adaptive filter $\hat{\mathbf{h}}(m)$ of length L can be partitioned into \mathcal{K} sub-filters each of length N .

In this chapter, the variable k is denoted as the block-index for $k = 0, 1, \dots, \mathcal{K} - 1$, where the $N \times 1$ vector

$$\hat{\mathbf{h}}_k(m) = [\hat{h}_{kN}(m) \ \hat{h}_{kN+1}(m) \ \dots \ \hat{h}_{kN+N-1}(m)]^T, \quad (5.23)$$

is the k^{th} sub-filter of $\hat{\mathbf{h}}(m)$ such that the relationship between the estimated impulse response $\hat{\mathbf{h}}(m)$ and each sub-filter $\hat{\mathbf{h}}_k(m)$ can be explicitly expressed by

$$\hat{\mathbf{h}}(m) = \left[\underbrace{\hat{h}_0(m) \ \dots \ \hat{h}_{N-1}(m)}_{\hat{\mathbf{h}}_0(m)} \ \dots \ \dots \ \dots \ \underbrace{\hat{h}_{L-N}(m) \ \dots \ \hat{h}_{L-1}(m)}_{\hat{\mathbf{h}}_{\mathcal{K}-1}(m)} \right]^T. \quad (5.24)$$

Defining a $N \times N$ Toeplitz matrix $\mathbb{T}(m - k)$ obtained from $\mathbf{X}^T(m)$ given by [11]

$$\mathbb{T}(m - k) = \begin{bmatrix} x(mN - kN) & x(mN - kN - 1) & \dots & x(mN - kN - N + 1) \\ x(mN - kN + 1) & x(mN - kN) & \dots & x(mN - kN - N + 2) \\ \vdots & \vdots & \ddots & \vdots \\ x(mN - kN + N - 1) & x(mN - kN + N - 2) & \dots & x(mN - kN) \end{bmatrix}_{N \times N}, \quad (5.25)$$

the time-domain filter output, obtained using convolution operation, can then be expressed in terms of the output of each sub-filter, using (5.20) and (5.25), as

$$\hat{\mathbf{y}}(m) = \sum_{k=0}^{\mathcal{K}-1} \mathbb{T}(m - k) \hat{\mathbf{h}}_k(m - 1). \quad (5.26)$$

To express the output in the frequency-domain, the Toeplitz matrix $\mathbb{T}(m-k)$ can be transformed to a circulant matrix [11] by first defining

$$\mathbb{T}'(m-k) = \begin{bmatrix} x(mN-kN+N) & x(mN-kN+N-1) & \cdots & x(mN-kN+1) \\ x(mN-kN-N+1) & x(mN-kN+N) & \cdots & x(mN-kN+2) \\ \vdots & \ddots & \ddots & \vdots \\ x(mN-kN-1) & x(mN-kN-2) & \cdots & x(mN-kN+N) \end{bmatrix}_{N \times N} \quad (5.27)$$

Using (5.25) and (5.27), the circulant matrix is then given as

$$\mathbb{C}(m-k) = \begin{bmatrix} \mathbb{T}'(m-k) & \mathbb{T}(m-k) \\ \mathbb{T}(m-k) & \mathbb{T}'(m-k) \end{bmatrix}_{2N \times 2N} \quad (5.28)$$

which can then be decomposed as

$$\mathbb{C}(m-k) = \mathbf{F}_{2N \times 2N}^{-1} \underline{\mathbf{D}}(m-k) \mathbf{F}_{2N \times 2N}, \quad (5.29)$$

where $\underline{\mathbf{D}}(m-k)$ is a $2N \times 2N$ diagonal matrix whose elements are the discrete Fourier transform of the first column of $\mathbb{C}(m-k)$ [118], i.e.,

$$\underline{\mathbf{D}}(m-k) = \text{diag} \left\{ \text{FFT} \left\{ x(mN-kN+N) \ x(mN-kN-N+1) \ \dots \ x(mN-kN+N-1) \right\} \right\}. \quad (5.30)$$

It should be noted that the diagonal of $\mathbb{C}(m-k)$ is arbitrary, but it is customary to set it equal to the first sample of the previous block [11]. This ensures the circulatory structure of $\mathbb{C}(m-k)$ which can be observed if $\mathbb{C}(m-k)$ is expanded explicitly (c.f. (5.36)).

It can be seen that for a single filter block $\mathcal{K} = 1$, block size $N = L$ and hence $\underline{\mathbf{D}}(m)$ is equivalent to $\underline{\mathbf{D}}(m)$ as defined in (5.7). To express the MDF update equation, the following quantities are first defined:

$$\underline{\mathbf{y}}(m) = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{y}(m), \quad (5.31)$$

$$\underline{\hat{\mathbf{h}}}_k(m) = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10} \hat{\mathbf{h}}_k(m), \quad (5.32)$$

$$\underline{\mathbf{e}}(m) = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{e}(m), \quad (5.33)$$

where $\mathbf{W}_{2N \times N}^{01}$ and $\mathbf{W}_{2N \times N}^{10}$ are given by (5.2a) and (5.2c) respectively. The k^{th} sub-filter of the MDF structure [23] is then updated by

$$\hat{\mathbf{h}}_k(m) = \hat{\mathbf{h}}_k(m-1) + 2\mu(1-\lambda)\mathbf{G}_{2N \times 2N}^{10}\mathbf{D}^*(m-k)[\mathcal{P}_{\text{MDF}}(m) + \delta_{\text{MDF}}\mathbf{I}_{2N \times 2N}]^{-1}\mathbf{e}(m), \quad (5.34)$$

for sub-filter block indices $k = 0, 1, \dots, \mathcal{K} - 1$, where $0 \ll \lambda < 1$ is the forgetting factor, $0 < \mu \leq 1$ is the adaptive step-size, δ_{MDF} is the regularization parameter, $\mathbf{G}_{2N \times 2N}^{10}$ is defined in (5.2k) and $\mathbf{I}_{2N \times 2N}$ is the $2N \times 2N$ identity matrix. The spectra estimation can be obtained recursively using the transformed input signal of the first sub-filter

$$\mathcal{P}_{\text{MDF}}(m) = \lambda\mathcal{P}_{\text{MDF}}(m-1) + (1-\lambda)\mathbf{D}^*(m)\mathbf{D}(m), \quad (5.35)$$

where the diagonal matrix $\mathcal{P}_{\text{MDF}}(m)$ is of dimension $2N \times 2N$. The MDF algorithm is summarized in Table 5.3 of Section 5.8.7.

5.3.3 General derivation of frequency-domain algorithms

One of the most recent developments of frequency-domain adaptive algorithms for AEC application is the derivation of the FLMS and MDF algorithms using a frequency-domain cost function as presented in [11]. In this section, this derivation is reviewed where a block recursive least-squares criterion using block size N independent of the adaptive filter length L is employed. As before, a noiseless case is considered without the loss of generality, for this frequency-domain derivation. Using (5.28), the circulant matrix $\mathbb{C}(m-k)$ defined in (5.28) can be expressed explicitly as

$$\mathbb{C}(m-k) = \begin{bmatrix} x(mN-kN+N) & \dots & x(mN-kN+1) & | & x(mN-kN) & \dots & x(mN-kN-N+1) \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ x(mN-kN-1) & \dots & x(mN-kN+N) & | & x(mN-kN+N-1) & \dots & x(mN-kN) \\ \hline x(mN-kN) & \dots & x(mN-kN-N+1) & | & x(mN-kN+N) & \dots & x(mN-kN+1) \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ x(mN-kN+N-1) & \dots & x(mN-kN) & | & x(mN-kN-1) & \dots & x(mN-kN+N) \end{bmatrix}_{2N \times 2N} \quad (5.36)$$

Defining the $2N \times 1$ filter output as

$$\hat{\mathbf{y}}'(m) = [\hat{y}(mN + N) \dots \hat{y}(mN - 1) \hat{y}(mN) \dots \hat{y}(mN + N - 1)]^T, \quad (5.37)$$

it can be shown, using (5.36), that

$$\hat{\mathbf{y}}'(m) = \sum_{k=0}^{\mathcal{K}-1} \mathbf{C}(m-k) \mathbf{W}_{2N \times N}^{10} \hat{\mathbf{h}}_k(m-1), \quad (5.38)$$

where the windowing matrix $\mathbf{W}_{2N \times N}^{10}$ and the k^{th} sub-filter $\hat{\mathbf{h}}_k(m-1)$ are defined in (5.2c) and (5.23) respectively. Using (5.38), the $N \times 1$ *a priori* block-error vector defined in (5.22) can be expressed as

$$\mathbf{e}(m) = \mathbf{y}(m) - \mathbf{W}_{N \times 2N}^{01} \hat{\mathbf{y}}'(m), \quad (5.39)$$

where $\mathbf{y}(m)$ is defined in (5.21) and the windowing matrix $\mathbf{W}_{N \times 2N}^{01}$, defined in (5.2b), selects the last N elements of $\hat{\mathbf{y}}'(m)$. To express the *a priori* error in the frequency-domain, the following quantities are first defined,

$$\underline{\hat{\mathbf{y}}}(m) = \mathbf{F}_{2N \times 2N} \hat{\mathbf{y}}'(m), \quad (5.40)$$

$$\underline{\mathbf{y}}(m) = \mathbf{F}_{N \times N} \mathbf{y}(m), \quad (5.41)$$

$$\underline{\mathbf{e}}(m) = \mathbf{F}_{N \times N} \mathbf{e}(m), \quad (5.42)$$

$$\underline{\hat{\mathcal{H}}}_k(m) = \mathbf{F}_{N \times N} \hat{\mathbf{h}}_k(m). \quad (5.43)$$

Employing (5.38), (5.29) and multiplying $\mathbf{F}_{N \times N}$ to (5.39), the frequency-domain *a*

priori error [11] can be expressed as

$$\begin{aligned}
\mathbf{F}_{N \times N} \mathbf{e}(m) &= \mathbf{F}_{N \times N} \mathbf{y}(m) - \mathbf{F}_{N \times N} \mathbf{W}_{N \times 2N}^{01} \hat{\mathbf{y}}'(m) \\
\underline{\mathbf{e}}(m) &= \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \hat{\underline{\mathbf{y}}}'(m) \\
&= \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \sum_{k=0}^{\mathcal{K}-1} \mathbf{F}_{2N \times 2N} \mathbf{C}(m-k) \mathbf{W}_{2N \times N}^{10} \mathbf{F}_{N \times N}^{-1} \hat{\underline{\mathbf{H}}}_k(m-1) \\
&= \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \sum_{k=0}^{\mathcal{K}-1} \underline{\mathbf{D}}(m-k) \mathbf{G}_{2N \times N}^{10} \hat{\underline{\mathbf{H}}}_k(m-1), \tag{5.44}
\end{aligned}$$

where $\mathbf{G}_{N \times 2N}^{01}$ and $\mathbf{G}_{2N \times N}^{10}$ are defined by (5.2i) and (5.2j) respectively. For compactness, the summation sign in (5.44) can be removed by first defining the $2N \times N$ matrix

$$\underline{\mathbf{u}}(m-k) = \underline{\mathbf{D}}(m-k) \mathbf{G}_{2N \times N}^{10} \tag{5.45}$$

such that when its offset versions are concatenated, a $2N \times L$ matrix $\underline{\mathbf{U}}(m)$ given by

$$\underline{\mathbf{U}}(m) = \begin{bmatrix} \underline{\mathbf{u}}(m) & \underline{\mathbf{u}}(m-1) & \dots & \underline{\mathbf{u}}(m-\mathcal{K}+1) \end{bmatrix}_{2N \times L} \tag{5.46}$$

is obtained. In a similar manner, sub-filters $\hat{\underline{\mathbf{H}}}_k(m)$, as defined by (5.43) for $k = 0, 1, \dots, \mathcal{K}-1$, can be concatenated, giving

$$\hat{\underline{\mathbf{H}}}^T(m) = \begin{bmatrix} \hat{\underline{\mathbf{H}}}_0^T(m) & \hat{\underline{\mathbf{H}}}_1^T(m) & \dots & \hat{\underline{\mathbf{H}}}_{\mathcal{K}-1}^T(m) \end{bmatrix}_{1 \times L} \tag{5.47}$$

The $N \times 1$ *a posteriori* error can be expressed, similar to (5.44), by

$$\underline{\mathbf{e}}_p(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \sum_{k=0}^{\mathcal{K}-1} \underline{\mathbf{D}}(m-k) \mathbf{G}_{2N \times N}^{10} \hat{\underline{\mathbf{H}}}_k(m), \tag{5.48}$$

from which, substituting (5.46) and (5.47) into (5.48) gives

$$\underline{\mathbf{e}}_p(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \underline{\mathbf{U}}(m) \hat{\underline{\mathbf{H}}}(m). \tag{5.49}$$

Defining the superscript H as the Hermitian operator, the time-averaged frequency-

domain cost function [11] [53] [123] can be defined as

$$\begin{aligned}
\underline{\mathcal{J}}_p(m) &= (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\boldsymbol{\varepsilon}}_p^H(r) \underline{\boldsymbol{\varepsilon}}_p(r) \\
&= (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \left[\underline{\boldsymbol{y}}^H(r) \underline{\boldsymbol{y}}(r) - \underline{\boldsymbol{y}}^H(r) \mathbf{G}_{N \times 2N}^{01} \underline{\boldsymbol{U}}(r) \hat{\underline{\boldsymbol{H}}}(r) \right. \\
&\quad \left. - \hat{\underline{\boldsymbol{H}}}^H(r) \underline{\boldsymbol{U}}^H(r) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\boldsymbol{y}}(r) \right. \\
&\quad \left. + \hat{\underline{\boldsymbol{H}}}^H(r) \underline{\boldsymbol{U}}^H(r) (\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} \underline{\boldsymbol{U}}(r) \hat{\underline{\boldsymbol{H}}}(r) \right]. \quad (5.50)
\end{aligned}$$

where $0 \ll \lambda < 1$ is the forgetting factor. As shown in Section 5.8.1, the last term $(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01}$ in (5.50) can be expressed as

$$(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} = 0.5 \times \mathbf{G}_{2N \times 2N}^{01}. \quad (5.51)$$

Similar to the time-domain stochastic gradient approach as discussed in Section 2.4, the gradient operator can be applied to the cost function $\underline{\mathcal{J}}_p(m)$ giving

$$\begin{aligned}
\nabla \underline{\mathcal{J}}_p(m) &= \frac{\partial \underline{\mathcal{J}}_p(m)}{\partial \hat{\underline{\boldsymbol{H}}}^H(m)} \\
&= -(1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\boldsymbol{U}}^H(r) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\boldsymbol{y}}(r) \\
&\quad + 0.5(1 - \lambda) \left[\sum_{r=0}^m \lambda^{m-r} \underline{\boldsymbol{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\boldsymbol{U}}(r) \right] \hat{\underline{\boldsymbol{H}}}(m). \quad (5.52)
\end{aligned}$$

As shown in Section 5.8.2, the first term can be simplified using

$$(\mathbf{G}_{N \times 2N}^{01})^H \underline{\boldsymbol{y}}(r) = 0.5 \times \underline{\boldsymbol{y}}(r), \quad (5.53)$$

where the $2N \times 1$ vector $\underline{\boldsymbol{y}}(r)$ is defined in (5.31). Equating (5.52) to zero and defining

$$\underline{\boldsymbol{s}}(m) = (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\boldsymbol{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\boldsymbol{U}}(r), \quad (5.54)$$

$$\underline{\boldsymbol{s}}(m) = (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\boldsymbol{U}}^H(r) \underline{\boldsymbol{y}}(r), \quad (5.55)$$

the expression

$$\underline{\mathbf{S}}(m)\widehat{\underline{\mathbf{H}}}(m) = \underline{\mathbf{s}}(m) \quad (5.56)$$

is obtained.

The significance of (5.56) can now be seen, which depicts a set of *normal equations* in the *frequency-domain* [11] where $\underline{\mathbf{S}}(m)$, of dimension $L \times L$, is the frequency-domain counterpart of the time-averaged autocorrelation matrix $\Psi(n)$ defined in (4.9). Similarly, the $L \times 1$ vector $\underline{\mathbf{s}}(m)$ is the frequency-domain counterpart of the time-averaged cross-correlation vector $\Theta(n)$ defined in (4.10). Similar to (2.62) and (2.63), the frequency-domain adaptive algorithm solves these normal equations by expressing $\underline{\mathbf{S}}(m)$ and $\underline{\mathbf{s}}(m)$ recursively. The term $\underline{\mathbf{S}}(m)$ can be expressed recursively [11] by first noting that

$$\begin{aligned} \underline{\mathbf{S}}(m-1) &= (1-\lambda) \sum_{r=0}^{m-1} \lambda^{m-1-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(r), \\ \lambda \underline{\mathbf{S}}(m-1) &= (1-\lambda) \sum_{r=0}^{m-1} \lambda^{m-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(r), \end{aligned} \quad (5.57)$$

hence giving

$$\begin{aligned} \underline{\mathbf{S}}(m) &= (1-\lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(r) \\ &= (1-\lambda) \left[\sum_{r=0}^{m-1} \lambda^{m-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(r) \right] + (1-\lambda) \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) \\ &= \lambda \underline{\mathbf{S}}(m-1) + (1-\lambda) \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m), \end{aligned} \quad (5.58)$$

where (5.57) is employed in the last step. In a similar manner, using the expressions

$$\begin{aligned} \underline{\mathbf{s}}(m-1) &= (1-\lambda) \sum_{r=0}^{m-1} \lambda^{m-1-r} \underline{\mathbf{U}}^H(r) \underline{\mathbf{y}}(r), \\ \lambda \underline{\mathbf{s}}(m-1) &= (1-\lambda) \sum_{r=0}^{m-1} \lambda^{m-r} \underline{\mathbf{U}}^H(r) \underline{\mathbf{y}}(r), \end{aligned} \quad (5.59)$$

the recursive formulation of the $L \times 1$ vector $\underline{\mathbf{s}}(m)$ can be achieved [11] giving

$$\begin{aligned}
\underline{\mathbf{s}}(m) &= (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) \underline{\mathbf{y}}(r) \\
&= (1 - \lambda) \left[\sum_{r=0}^{m-1} \lambda^{m-1-r} \underline{\mathbf{U}}^H(r) \underline{\mathbf{y}}(r) \right] + (1 - \lambda) \underline{\mathbf{U}}^H(m) \underline{\mathbf{y}}(m) \\
&= \lambda \underline{\mathbf{s}}(m-1) + (1 - \lambda) \underline{\mathbf{U}}^H(m) \underline{\mathbf{y}}(m) .
\end{aligned} \tag{5.60}$$

Employing (5.58) and (5.60), the frequency-domain normal equations in (5.56) can be simplified as

$$\begin{aligned}
\hat{\underline{\mathbf{H}}}(m) &= \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{s}}(m) \\
&= \underline{\mathbf{S}}^{-1}(m) \left[\lambda \underline{\mathbf{s}}(m-1) + (1 - \lambda) \underline{\mathbf{U}}^H(m) \underline{\mathbf{y}}(m) \right] \\
&= \lambda \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{S}}(m-1) \hat{\underline{\mathbf{H}}}(m-1) + (1 - \lambda) \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{U}}^H(m) \underline{\mathbf{y}}(m) \\
&= \underline{\mathbf{S}}^{-1}(m) \left[\underline{\mathbf{S}}(m) - (1 - \lambda) \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) \right] \hat{\underline{\mathbf{H}}}(m-1) \\
&\quad + (1 - \lambda) \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{U}}^H(m) \underline{\mathbf{y}}(m) \\
&= \hat{\underline{\mathbf{H}}}(m-1) - (1 - \lambda) \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{U}}^H(m) \left[\mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) \hat{\underline{\mathbf{H}}}(m-1) - \underline{\mathbf{y}}(m) \right] ,
\end{aligned} \tag{5.61}$$

which describes a frequency-domain recursive filter update. Similar to (5.49), the $N \times 1$ *a priori* error can be expressed as

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{N \times 2N}^{01} \underline{\mathbf{U}}(m) \hat{\underline{\mathbf{H}}}(m-1) , \tag{5.62}$$

such that when pre-multiplied by $(\mathbf{G}_{N \times 2N}^{01})^H$ and using the relation

$$(\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{e}}(m) = 0.5 \times \underline{\mathbf{e}}(m) , \tag{5.63}$$

the $2N \times 1$ vector

$$\begin{aligned}
0.5 \underline{\mathbf{e}}(m) &= (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(m) - (\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} \underline{\mathbf{U}}(m) \hat{\underline{\mathbf{H}}}(m-1) \\
&= 0.5 \underline{\mathbf{y}}(m) - 0.5 \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) \hat{\underline{\mathbf{H}}}(m-1)
\end{aligned} \tag{5.64}$$

is obtained. The frequency-domain update equation for $\hat{\mathbf{H}}(m)$ given in (5.61) can now be expressed as

$$\hat{\mathbf{H}}(m) = \hat{\mathbf{H}}(m-1) + (1-\lambda)\mathbf{S}^{-1}(m)\mathbf{U}^H(m)\mathbf{e}(m). \quad (5.65)$$

This update equation can be further simplified by defining [11] the concatenated matrices

$$\mathbf{G}_{2L \times L}^{10} = \text{diag}\left\{ \underbrace{\mathbf{G}_{2N \times N}^{10} \mathbf{G}_{2N \times N}^{10} \cdots \mathbf{G}_{2N \times N}^{10}}_{\mathcal{K} \text{ sub-matrices}} \right\}, \quad (5.66)$$

$$\mathbf{D}(m) = [\mathbf{D}(m) \mathbf{D}(m-1) \cdots \mathbf{D}(m-\mathcal{K}+1)]_{2N \times 2L}, \quad (5.67)$$

where $\mathbf{D}(m-k)$, for $k=0, 1, \dots, \mathcal{K}-1$, are the diagonal matrices each having dimension $2N \times 2N$ as defined in (5.30). Similar to (5.45), the $2N \times L$ matrix $\mathbf{U}(m)$ defined in (5.46) can be expressed in terms of $\mathbf{D}(m)$ given as

$$\mathbf{U}(m) = \mathbf{D}(m)\mathbf{G}_{2L \times L}^{10}, \quad (5.68)$$

from which the $L \times L$ matrix $\mathbf{S}(m)$ in (5.58) and the frequency-domain block error $\mathbf{e}(m)$ in (5.64) can now be expressed respectively as

$$\mathbf{S}(m) = \lambda\mathbf{S}(m-1) + (1-\lambda)(\mathbf{G}_{2L \times L}^{10})^H \mathbf{D}^H(m) \mathbf{G}_{2N \times 2N}^{01} \mathbf{D}(m) \mathbf{G}_{2L \times L}^{10}, \quad (5.69)$$

$$\mathbf{e}(m) = \mathbf{y}(m) - \mathbf{G}_{2N \times 2N}^{01} \mathbf{D}(m) \mathbf{G}_{2L \times L}^{10} \hat{\mathbf{H}}(m-1). \quad (5.70)$$

Noting that

$$\begin{aligned} \mathbf{G}_{2L \times L}^{10} \hat{\mathbf{H}}(m) &= \begin{bmatrix} \mathbf{G}_{2N \times N}^{10} & 0 & \cdots & 0 \\ 0 & \mathbf{G}_{2N \times N}^{10} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{G}_{2N \times N}^{10} \end{bmatrix}_{2L \times L} \begin{bmatrix} \hat{\mathbf{H}}_0(m) \\ \hat{\mathbf{H}}_1(m) \\ \vdots \\ \hat{\mathbf{H}}_{\mathcal{K}-1}(m) \end{bmatrix}_{L \times 1} \\ &= \left[\mathbf{G}_{2N \times N}^{10} \hat{\mathbf{H}}_0(m) \quad \mathbf{G}_{2N \times N}^{10} \hat{\mathbf{H}}_1(m) \quad \cdots \quad \mathbf{G}_{2N \times N}^{10} \hat{\mathbf{H}}_{\mathcal{K}-1}(m) \right]^T, \quad (5.71) \end{aligned}$$

such that for each sub-filter, $\mathbf{G}_{2N \times N}^{01} \hat{\mathbf{H}}_k(m)$ can be expressed for $k = 0, 1, \dots, \mathcal{K} - 1$ as

$$\begin{aligned} \mathbf{G}_{2N \times N}^{10} \hat{\mathbf{H}}_k(m) &= \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10} \mathbf{F}_{N \times N}^{-1} \hat{\mathbf{H}}_k(m) \\ &= \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{10} \hat{\mathbf{h}}_k(m) \\ &= \hat{\mathbf{h}}_k(m), \end{aligned} \quad (5.72)$$

where the definitions of (5.32) and (5.43) have been employed. Employing (5.72), the relationship between the $2L \times 1$ estimated impulse response $\hat{\mathbf{h}}(m)$ given by (5.8) and $\hat{\mathbf{H}}(m)$ defined by (5.47) is established:

$$\begin{aligned} \mathbf{G}_{2L \times L}^{10} \hat{\mathbf{H}}(m) &= \left[\hat{\mathbf{h}}_0^T(m) \hat{\mathbf{h}}_1^T(m) \dots \hat{\mathbf{h}}_{\mathcal{K}-1}^T(m) \right]^T \\ &= \hat{\mathbf{h}}(m). \end{aligned} \quad (5.73)$$

The frequency-domain *a priori* error $\underline{\mathbf{e}}(m)$ in (5.70) can then be expressed as

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}_{2N \times 2N}^{01} \underline{\mathcal{D}}(m) \hat{\mathbf{h}}(m-1). \quad (5.74)$$

Premultiplying (5.65) by $\mathbf{G}_{2L \times L}^{10}$ and using (5.68) and (5.73), the update equation can be expressed by

$$\hat{\mathbf{h}}(m) = \hat{\mathbf{h}}(m-1) + (1-\lambda) \mathbf{G}_{2L \times L}^{10} \underline{\mathbf{S}}^{-1}(m) (\mathbf{G}_{2L \times L}^{10})^H \underline{\mathcal{D}}^H(m) \underline{\mathbf{e}}(m). \quad (5.75)$$

To simplify this update equation further, the $L \times L$ matrix $\underline{\mathbf{S}}(m)$ defined by (5.58), can be expressed in terms of the $2N \times 2L$ matrix $\underline{\mathcal{D}}(m)$ given in (5.67), by first defining a $2L \times 2L$ matrix [11]

$$\begin{aligned} \underline{\mathbb{B}}(m) &= (1-\lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathcal{D}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathcal{D}}(r) \\ &= \lambda \underline{\mathbb{B}}(m-1) + (1-\lambda) \underline{\mathcal{D}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathcal{D}}(m) \end{aligned} \quad (5.76)$$

where, as shown in Section 5.8.3, the relationship between $\underline{\mathbf{S}}(m)$ and $\underline{\mathbb{B}}(m)$ is given by

$$\underline{\mathbf{S}}(m) = (\mathbf{G}_{2L \times L}^{10})^H \underline{\mathbb{B}}(m) \mathbf{G}_{2L \times L}^{10}. \quad (5.77)$$

As shown further in Section 5.8.4, the term $\mathbf{G}_{2L \times L}^{10} \underline{\mathbf{S}}^{-1}(m) (\mathbf{G}_{2L \times L}^{10})^H$ in (5.75) can be simplified giving

$$\mathbf{G}_{2L \times L}^{10} \underline{\mathbf{S}}^{-1}(m) (\mathbf{G}_{2L \times L}^{10})^H = \mathbf{G}_{2L \times 2L}^{10} \underline{\mathbf{B}}^{-1}(m) \quad (5.78)$$

and hence the frequency-domain update equation in (5.75) can now be expressed as

$$\hat{\mathbf{h}}(m) = \hat{\mathbf{h}}(m-1) + (1-\lambda) \mathbf{G}_{2L \times 2L}^{10} \underline{\mathbf{B}}^{-1}(m) \underline{\mathbf{D}}^H(m) \underline{\mathbf{e}}(m), \quad (5.79)$$

while $\mathbf{G}_{2L \times 2L}^{10}$ is defined by (5.2g).

Thus far, the frequency-domain algorithm has been defined by (5.74), (5.76) and (5.79). It should be noted that since $\underline{\mathbf{B}}(m)$ is not diagonal, the computation of $\underline{\mathbf{B}}^{-1}(m)$ in (5.79) is not computationally feasible. It has been shown and discussed in detail [11] that one can approximate $\mathbf{G}_{2N \times 2N}^{01}$ with an identity matrix scaled by a factor of 0.5, i.e.,

$$\mathbf{G}_{2N \times 2N}^{01} \approx \mathbf{I}_{2N \times 2N} / 2. \quad (5.80)$$

Using this approximation, the recursive computation of $\underline{\mathbf{B}}(m)$ and update of $\hat{\mathbf{h}}(m)$ is respectively given by

$$\underline{\mathbf{B}}'(m) = \lambda \underline{\mathbf{B}}'(m-1) + (1-\lambda) \underline{\mathbf{D}}^H(m) \underline{\mathbf{D}}(m) \quad (5.81)$$

$$\hat{\mathbf{h}}(m) = \hat{\mathbf{h}}(m-1) + 2\mu(1-\lambda) \mathbf{G}_{2L \times 2L}^{10} [\underline{\mathbf{B}}'(m) + \delta \mathbf{I}_{2L \times 2L}]^{-1} \underline{\mathbf{D}}^H(m) \underline{\mathbf{e}}(m), \quad (5.82)$$

where the *a priori* error $\underline{\mathbf{e}}(m)$ is defined by (5.74). Note that the factor of 0.5 has been absorbed into the adaptive step-size $0 < \mu \leq 1$ while δ is the regularization parameter [11] and $\mathbf{I}_{2L \times 2L}$ is the $2L \times 2L$ identity matrix.

It is now apparent that the minimization of the time-averaged cost function $\underline{\mathcal{J}}_p(m)$ in (5.50) arrives at a generalized frequency-domain adaptive algorithm with an update equation given by (5.82). More importantly, the link between the generalized frequency-domain equations governed by (5.81) and (5.82) and the FLMS algorithm as described in Section 5.3.1 can be seen. For a single block, $\mathcal{K} = 1$ giving block size $N = L$, the $2N \times 2L$ matrix $\underline{\mathbf{D}}(m)$ defined by (5.67) is equivalent to the $2L \times 2L$ diagonal matrix $\underline{\mathbf{D}}(m)$ defined

by (5.7). Consequently, the $2L \times 2L$ matrix $\mathbb{B}'(m)$ given in (5.81) is equivalent to the $2L \times 2L$ matrix $\mathcal{P}_{\text{FLMS}}(m)$ defined by (5.14). As a result, (5.81) and (5.82) are equivalent to (5.14) and (5.17) respectively giving the FLMS algorithm for block size $N = L$ (i.e., $\mathcal{K} = 1$).

Following the approach presented in [11], the link between (5.81), (5.82) and the MDF algorithm can be established using the following approximation

$$\mathbb{B}'(m) \approx \text{diag}\left\{ \underbrace{\underline{\mathcal{S}}(m) \dots \underline{\mathcal{S}}(m)}_{\mathcal{K} \text{ sub-matrices}} \right\} \quad (5.83)$$

where, using (5.67), each sub-matrix

$$\underline{\mathcal{S}}(m) = \lambda \underline{\mathcal{S}}(m-1) + (1-\lambda) \underline{\mathbf{D}}^*(m) \underline{\mathbf{D}}(m) \quad (5.84)$$

is diagonal, while $\underline{\mathbf{D}}(m)$ is a $2N \times 2N$ diagonal matrix defined by (5.30) for $k = 0$. With this approximation, the $2N \times 2N$ diagonal matrix $\underline{\mathcal{S}}(m)$ defined by (5.84) is equivalent to the $2N \times 2N$ matrix $\mathcal{P}_{\text{MDF}}(m)$ defined by (5.35). In addition, the $2L \times 1$ frequency-domain estimated impulse response $\hat{\mathbf{h}}(m)$ defined in (5.82) comprises of \mathcal{K} sub-filters each having dimension $2N \times 1$ as can be seen from (5.73). These sub-filters are each updated using (5.34) hence obtaining the MDF algorithm.

As a final comment, matrices $\mathbf{G}_{2L \times 2L}^{10}$ and $\mathbf{G}_{2N \times 2N}^{10}$ in (5.17), (5.34) and (5.82) form a constrain to the adaptive algorithm such that with the overlap-save FFT computation, linear convolution can be achieved. By approximating these matrices with $\mathbf{I}_{2L \times 2L}/2$ and $\mathbf{I}_{2N \times 2N}/2$ respectively, unconstrained algorithms can be obtained. Although these unconstrained algorithms, such as the UFLMS algorithm [109], require three FFT blocks compared to five for the constrained algorithms, their performances in terms of convergence rate and misalignment are generally degraded due to the effects of circular convolution [124] [125].

5.3.4 Steady-state misalignment

Following the approach in [11], the steady-state misalignment for the single channel frequency-domain algorithm presented in Section 5.3.3 is reviewed. The frequency-domain system mismatch vector, of dimension $L \times 1$, is defined as

$$\underline{\mathbf{V}}(m) = \underline{\mathbf{H}} - \widehat{\underline{\mathbf{H}}}(m), \quad (5.85)$$

where $\underline{\mathbf{H}}$ is the frequency-domain impulse response of the unknown system for which has been assumed to be time-invariant. The frequency-domain noise sequence

$$\underline{\mathbf{w}}'(m) = \mathbf{F}_{2N \times 2N} \mathbf{w}'(m) \quad (5.86)$$

is further defined, where the $2N \times 1$ vector $\mathbf{w}'(m)$ is given, similar to $\widehat{\mathbf{y}}'(m)$ in (5.37); as

$$\mathbf{w}'(m) = [w(mN + N) \dots w(mN - 1) w(mN) \dots w(mN + N - 1)]^T. \quad (5.87)$$

The system mismatch vector $\underline{\mathbf{V}}(m)$ can be expressed recursively by employing (5.85), (5.53), (5.55) and (5.56) and first expressing

$$\begin{aligned} \underline{\mathbf{V}}(m) &= \underline{\mathbf{H}} - \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{s}}(m) \\ &= \underline{\mathbf{H}} - 2\underline{\mathbf{S}}^{-1}(m)(1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) (\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(r) \\ &= \underline{\mathbf{H}} - 2\underline{\mathbf{S}}^{-1}(m)(1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) (\mathbf{G}_{N \times 2N}^{01})^H \left(\mathbf{G}_{N \times 2N}^{01} \underline{\mathbf{U}}(r) \underline{\mathbf{H}} + \mathbf{G}_{N \times 2N}^{01} \mathbf{w}'(r) \right) \\ &= \underline{\mathbf{H}} - \underline{\mathbf{S}}^{-1}(m) \underline{\mathbf{S}}(m) \underline{\mathbf{H}} - \underline{\mathbf{S}}^{-1}(m)(1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \mathbf{w}'(r), \end{aligned} \quad (5.88)$$

where the relationship $(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} = 0.5 \times \mathbf{G}_{2N \times 2N}^{01}$ as shown in Section 5.8.1 is used. Employing the mean ergodic theorem [85] where statistical mean can be approximated by time averaging for large m , it can be assumed that [11]

$$\underline{\mathbf{S}}(m) \approx \underline{\mathbf{S}} = E \left\{ \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) \right\}, \quad (5.89)$$

where $E\{\cdot\}$ is defined as the mathematical expectation. Following the recursive approach as shown in (5.60), the recursive form of the system mismatch vector $\underline{\mathbf{V}}(m)$ is then given by

$$\begin{aligned}\underline{\mathbf{V}}(m) &= -(1-\lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{S}}^{-1} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{w}}(r) \\ &= \lambda \underline{\mathbf{V}}(m-1) - (1-\lambda) \underline{\mathbf{S}}^{-1} \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{w}}(m).\end{aligned}\quad (5.90)$$

Employing the independence assumption [35] and assuming that for large m , similar to as discussed in Section 3.3, the adaptive algorithm is able to track the unknown system such that

$$\mathbf{R}_{\underline{\mathbf{V}}} = E\{\underline{\mathbf{V}}(m) \underline{\mathbf{V}}^H(m)\} \approx E\{\underline{\mathbf{V}}(m-1) \underline{\mathbf{V}}^H(m-1)\}, \quad (5.91)$$

the relationship

$$\begin{aligned}E\{\underline{\mathbf{V}}(m) \underline{\mathbf{V}}^H(m)\} &\approx \lambda^2 \text{tr} E\{\underline{\mathbf{V}}(m-1) \underline{\mathbf{V}}^H(m-1)\} \\ &\quad + (1-\lambda)^2 E\{\underline{\mathbf{S}}^{-1} \underline{\mathbf{U}}^H(m) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(m) (\underline{\mathbf{S}}^{-1})^H\} \times \\ &\quad E\{\underline{\mathbf{w}}(m) \underline{\mathbf{w}}^H(m)\} \\ (1-\lambda^2) \mathbf{R}_{\underline{\mathbf{V}}} &= (1-\lambda)^2 \underline{\mathbf{S}}^{-1} \mathbf{R}_{\underline{\mathbf{w}}}\end{aligned}\quad (5.92)$$

is obtained, where $\mathbf{R}_{\underline{\mathbf{w}}} = E\{\underline{\mathbf{w}}(m) \underline{\mathbf{w}}^H(m)\}$ is the noise autocorrelation matrix while $\mathbf{G}_{2N \times 2N}^{01} (\mathbf{G}_{2N \times 2N}^{01})^H = \mathbf{G}_{2N \times 2N}^{01}$ and noting that $\underline{\mathbf{S}}$ is a Hermitian matrix, the relationship $\underline{\mathbf{S}}^{-1} \underline{\mathbf{S}} (\underline{\mathbf{S}}^{-1})^H = \underline{\mathbf{S}}^{-1} (\underline{\mathbf{S}}^{-1} \underline{\mathbf{S}}^H)^H = \underline{\mathbf{S}}^{-1}$ is employed. With the assumption that $(1-\lambda^2) \approx 2(1-\lambda)$ for $\lambda \approx 1$, the steady-state misalignment $\underline{\eta}' = \text{tr}\{\mathbf{R}_{\underline{\mathbf{V}}}\}$ is given by [11]

$$\begin{aligned}2(1-\lambda) \text{tr}\{\mathbf{R}_{\underline{\mathbf{V}}}\} &\approx (1-\lambda)^2 \text{tr}\{\underline{\mathbf{S}}^{-1}\} \sigma_w^2 \\ \underline{\eta}' &= \frac{1-\lambda}{2} \text{tr}\{\underline{\mathbf{S}}^{-1}\} \sigma_w^2.\end{aligned}\quad (5.93)$$

It can be seen that the steady-state misalignment is a reducing function of the forgetting factor λ and is proportional to the noise variance.

5.4 Effect of interchannel coherence on the conditioning of \mathbf{R}_{xx} for SAEC

Having reviewed frequency-domain adaptive algorithms for single channel AEC, the relationship between interchannel coherence and the conditioning of the two-channel input autocorrelation matrix \mathbf{R}_{xx} as defined in (4.24) is established for stereophonic acoustic echo cancellation (SAEC).

As discussed in Chapter 4, a serious problem encountered in SAEC is the existence of non-unique solutions [52]. It has been shown [90] [52] that for a practical stereophonic system, \mathbf{R}_{xx} is highly ill-conditioned. This is due to the high coherence between the two input signals $x_1(n)$ and $x_2(n)$, as depicted in Fig. 4.1, which in turn degrades the misalignment performance of adaptive algorithms. For a single channel case, performance of adaptive algorithms in terms of their final misalignment is affected by the conditioning of the input autocorrelation matrix [111]. In a stereophonic case however, although it has generally been noted that the conditioning of \mathbf{R}_{xx} is degraded by the high interchannel coherence between $x_1(n)$ and $x_2(n)$, no explicit relationship between the two has been established. The aim of this section is to establish this relationship which then allows one to gain an insight of how interchannel coherence degrades the steady-state misalignment performance of SAEC algorithms through the ill-conditioning of \mathbf{R}_{xx} . As will be discussed, this relationship can be achieved by first decomposing \mathbf{R}_{xx} using frequency-domain quantities and exploiting the E-norm condition number [111]. Using this relationship, one can determine the level of ill-conditioning of \mathbf{R}_{xx} through the interchannel coherence estimate and design regularization parameters so as to improve the conditioning of \mathbf{R}_{xx} hence giving good misalignment performance such as shown in [113]. The validity of the established relationship will be verified for both white Gaussian noise (WGN) and speech input signals, showing how the condition number is affected by the interchannel coherence which in turn affects the performance of a two-channel frequency-domain adaptive algorithm [11] in terms of its steady-state misalignment.

5.4.1 Two-channel autocorrelation matrix and the normal equations

With reference to Fig. 4.1, the j^{th} channel tap-input vector $\mathbf{x}_j(n)$ (assumed to be zero mean) is defined in (4.3) and is reproduced here for convenience

$$\mathbf{x}_j(n) = [x_j(n) \ x_j(n-1) \ \dots \ x_j(n-L+1)]^T \quad (5.94)$$

for $j = 1, 2$, where L and the superscript T are the length of the adaptive filter and vector transposition respectively. Let

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n) \ \mathbf{x}_2^T(n)]^T \quad (5.95)$$

be the two-channel concatenated tap-input vector which then gives the two-channel correlation matrix as defined by (4.24) and is reproduced here for convenience

$$\begin{aligned} \mathbf{R}_{\mathbf{xx}} &= E\{\mathbf{x}(n)\mathbf{x}^T(n)\} \\ &= \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix}_{2L \times 2L} \end{aligned} \quad (5.96)$$

It had been noted and discussed in Chapter 2 that adaptive algorithms aim to solve the normal equations [35] given by

$$\hat{\mathbf{h}} = \mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{p} \ , \quad (5.97)$$

where $\mathbf{p} = E\{\mathbf{x}(n)y(n)\}$ is defined as the cross-correlation vector and $\hat{\mathbf{h}} = [\hat{\mathbf{h}}_1^T \ \hat{\mathbf{h}}_2^T]^T$ is the concatenated filter coefficients of channels 1 and 2. It is evident from (5.97) that an ill-conditioned $\mathbf{R}_{\mathbf{xx}}$ will yield a bad estimate of $\hat{\mathbf{h}}$ if determined by typical adaptive algorithms. The performance of adaptive algorithms in SAEC is further degraded by the interchannel coherence between $x_1(n)$ and $x_2(n)$ as will be shown below.

5.4.2 Autocorrelation matrix and spectral content

To establish the link between interchannel coherence and condition number of $\mathbf{R}_{\mathbf{xx}}$, the autocorrelation matrix can first be expressed in terms of its auto- and cross-spectral content

5.4 Effect of interchannel coherence on the conditioning of \mathbf{R}_{xx} for SAEC 170

and then exploiting the E-norm condition number [111]. Noting that for $L \rightarrow \infty$, a Toeplitz matrix is asymptotically equivalent to a circulant matrix if its elements are absolutely summable [118], the $L \times L$ Toeplitz correlation matrix between the j^{th} and u^{th} channel \mathbf{R}_{ju} , given in (5.96) can be expressed as [11]

$$\mathbf{R}_{ju} = \mathbf{F}_{L \times L}^{-1} \underline{\mathbf{S}}_{ju} \mathbf{F}_{L \times L}, \quad (5.98)$$

for $j, u = 1, 2$, where $\mathbf{F}_{L \times L}$ is the $L \times L$ Fourier matrix. The $L \times L$ diagonal matrix

$$\underline{\mathbf{S}}_{ju} = \text{diag}\{S_{ju}(0) \ S_{ju}(1) \ \dots \ S_{ju}(L-1)\} \quad (5.99)$$

contains elements corresponding to the L frequency bins which are formed from the discrete Fourier transform (DFT) of the first column of \mathbf{R}_{ju} . Letting $r_{ju}(l)$ be the auto- and cross-correlation coefficients for $j = u$ and $j \neq u$ respectively, the spectral content between two signals is related to the correlation function [106] by

$$S_{ju}(f) = \sum_{l=-\infty}^{\infty} r_{ju}(l) e^{-i2\pi fl}, \quad f = 0, 1, \dots, L-1. \quad (5.100)$$

Using (5.98), \mathbf{R}_{xx} can be expressed in terms of its spectra as

$$\begin{aligned} \mathbf{R}_{xx} &= \begin{bmatrix} \mathbf{F}_{L \times L}^{-1} \underline{\mathbf{S}}_{11} \mathbf{F}_{L \times L} & \mathbf{F}_{L \times L}^{-1} \underline{\mathbf{S}}_{12} \mathbf{F}_{L \times L} \\ \mathbf{F}_{L \times L}^{-1} \underline{\mathbf{S}}_{21} \mathbf{F}_{L \times L} & \mathbf{F}_{L \times L}^{-1} \underline{\mathbf{S}}_{22} \mathbf{F}_{L \times L} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}_{L \times L}^{-1} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{F}_{L \times L}^{-1} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{S}}_{11} & \underline{\mathbf{S}}_{12} \\ \underline{\mathbf{S}}_{21} & \underline{\mathbf{S}}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{L \times L} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{F}_{L \times L} \end{bmatrix}, \quad (5.101) \end{aligned}$$

where $\mathbf{0}_{L \times L}$ is a null matrix of dimension $L \times L$.

5.4.3 The E-norm condition number of autocorrelation matrix

The condition number $\chi[\mathbf{A}]$ of a $2L \times 2L$ matrix \mathbf{A} is commonly computed using the l_2 -norm [126] and is denoted by

$$\begin{aligned}\chi_2[\mathbf{A}] &= \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 \\ &= \bar{\lambda}_{2L-1} / \bar{\lambda}_0,\end{aligned}\tag{5.102}$$

where $\|\cdot\|_2$ is the l_2 -norm operator and $\bar{\lambda}_l$ is the l^{th} eigenvalue of the positive definite matrix \mathbf{A} where $0 \leq \bar{\lambda}_0 \leq \bar{\lambda}_1 \leq \dots \leq \bar{\lambda}_{2L-1}$. However, it has been shown that the E-norm [111] is suitable for this current SAEC application as will be explained briefly in this section. The symmetric and positive definite correlation matrix can be diagonalized as

$$\mathbf{Q}^T \mathbf{R}_{xx} \mathbf{Q} = \bar{\Lambda},\tag{5.103}$$

where \mathbf{Q} is a unitary matrix such that $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$ and

$$\bar{\Lambda} = \text{diag}\{\bar{\lambda}_0 \ \bar{\lambda}_1 \ \dots \ \bar{\lambda}_{2L-1}\}\tag{5.104}$$

contains the eigenvalues of \mathbf{R}_{xx} with $0 \leq \bar{\lambda}_0 \leq \bar{\lambda}_1 \leq \dots \leq \bar{\lambda}_{2L-1}$. By definition, the square-root of \mathbf{R}_{xx} is given by [127]

$$\mathbf{R}_{xx}^{1/2} = \mathbf{Q} \bar{\Lambda}^{1/2} \mathbf{Q}^T.\tag{5.105}$$

Defining $\text{tr}\{\cdot\}$ as the trace operator, the E-norm of the $2L \times 2L$ matrix \mathbf{R}_{xx} is then defined as [111]

$$\|\mathbf{R}_{xx}\|_E = \left[\frac{1}{2L} \text{tr}\{\mathbf{R}_{xx}^T \mathbf{R}_{xx}\} \right]^{1/2}.\tag{5.106}$$

Noting that the squared Frobenius-norm [126] of \mathbf{R}_{xx} is defined as

$$\|\mathbf{R}_{xx}\|_F^2 = \text{tr}\{\mathbf{R}_{xx}^T \mathbf{R}_{xx}\},\tag{5.107}$$

the E-norm is then equivalent to the F-norm scaled by a factor $1/\sqrt{2L}$. Using (5.105), it follows that

$$\|\mathbf{R}_{xx}^{1/2}\|_E = \left[\frac{1}{2L} \text{tr}\{\mathbf{R}_{xx}\} \right]^{1/2}, \quad (5.108a)$$

$$\|\mathbf{R}_{xx}^{-1/2}\|_E = \left[\frac{1}{2L} \text{tr}\{\mathbf{R}_{xx}^{-1}\} \right]^{1/2}, \quad (5.108b)$$

which results in the E-norm condition number

$$\chi_E[\mathbf{R}_{xx}^{1/2}] = \|\mathbf{R}_{xx}^{1/2}\|_E \|\mathbf{R}_{xx}^{-1/2}\|_E. \quad (5.109)$$

Hence the E-norm of an identity matrix is one and if $\chi_E[\mathbf{R}_{xx}^{1/2}]$ is large, the correlation matrix \mathbf{R}_{xx} is said to be ill-conditioned. In addition, in order to study only the effect of interchannel coherence on the condition number, the factor $1/(2L)$ removes the dependency of the E-norm condition number on L hence making $\chi_E[\mathbf{R}_{xx}^{1/2}]$ a more suitable measure¹ than $\chi_F[\mathbf{R}_{xx}^{1/2}]$ where the latter is computed using $\|\mathbf{R}_{xx}^{1/2}\|_F = [\text{tr}\{\mathbf{R}_{xx}\}]^{1/2}$. It is further shown in [111] that $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ is a good measure of the conditioning of \mathbf{R}_{xx} .

5.4.4 Relationship between interchannel coherence and the conditioning of \mathbf{R}_{xx}

To compute $\chi_E[\mathbf{R}_{xx}^{1/2}]$, a matrix $\underline{\mathbf{S}}$ containing the auto- and cross-energy density spectra is first defined as

$$\underline{\mathbf{S}} = \begin{bmatrix} \underline{\mathbf{S}}_{11} & \underline{\mathbf{S}}_{12} \\ \underline{\mathbf{S}}_{21} & \underline{\mathbf{S}}_{22} \end{bmatrix}_{2L \times 2L} \quad (5.110)$$

such that $\text{tr}\{\mathbf{R}_{xx}\}$ from (5.101) can be computed using the following

$$\begin{aligned} \text{tr}\{\mathbf{R}_{xx}\} &= \text{tr}\left\{ \check{\mathbf{F}}_{2L \times 2L}^{-1} \underline{\mathbf{S}} \check{\mathbf{F}}_{2L \times 2L} \right\} \\ &= \sum_{l=0}^{L-1} [S_{11}(l) + S_{22}(l)], \end{aligned} \quad (5.111)$$

¹The dependency of $\chi_F[\mathbf{R}_{xx}^{1/2}]$ on L and its explicit relationship with $\chi_E[\mathbf{R}_{xx}^{1/2}]$ for WGN uncorrelated sequences $x_1(n)$ and $x_2(n)$ is further shown in Section 5.8.5.

where the $2L \times 2L$ matrices $\check{\mathbf{F}}_{2L \times 2L}$ and $\check{\mathbf{F}}_{2L \times 2L}^{-1}$ are defined respectively as

$$\check{\mathbf{F}}_{2L \times 2L} = \begin{bmatrix} \mathbf{F}_{L \times L} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{F}_{L \times L} \end{bmatrix}_{2L \times 2L}, \quad (5.112a)$$

$$\check{\mathbf{F}}_{2L \times 2L}^{-1} = \begin{bmatrix} \mathbf{F}_{L \times L}^{-1} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \mathbf{F}_{L \times L}^{-1} \end{bmatrix}_{2L \times 2L} \quad (5.112b)$$

and the relation $\text{tr}\{\mathbf{AB}\} = \text{tr}\{\mathbf{BA}\}$ is employed.

Using (5.112a) and (5.112b), $\text{tr}\{\mathbf{R}_{xx}^{-1}\}$ can be first simplified by

$$\text{tr}\{\mathbf{R}_{xx}^{-1}\} = \text{tr}\{\check{\mathbf{F}}_{2L \times 2L}^{-1} \underline{\mathbf{S}}^{-1} \check{\mathbf{F}}_{2L \times 2L}\} = \text{tr}\{\underline{\mathbf{S}}^{-1}\}. \quad (5.113)$$

Note that for the trivial case of $x_1(n) = x_2(n)$, \mathbf{R}_{xx}^{-1} does not exist. Using a similar approach to [11] [114], provided that the coherence is not equal to unity for any frequency, the $2L \times 2L$ inverse matrix $\underline{\mathbf{S}}^{-1}$ can be expressed as

$$\underline{\mathbf{S}}^{-1} = \begin{bmatrix} \underline{\mathbf{S}}_1^{-1} & \mathbf{0}_{L \times L} \\ \mathbf{0}_{L \times L} & \underline{\mathbf{S}}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{L \times L} & -\underline{\mathbf{S}}_{12} \underline{\mathbf{S}}_{22}^{-1} \\ -\underline{\mathbf{S}}_{21} \underline{\mathbf{S}}_{11}^{-1} & \mathbf{I}_{L \times L} \end{bmatrix}, \quad (5.114)$$

where $\mathbf{I}_{L \times L}$ is an $L \times L$ identity matrix and the sub-matrices

$$\underline{\mathbf{S}}_1 = \left[\mathbf{I}_{L \times L} - \underline{\mathbf{S}}_{12}^2 (\underline{\mathbf{S}}_{11}^{-1} \underline{\mathbf{S}}_{22}^{-1}) \right] \underline{\mathbf{S}}_{11}, \quad (5.115a)$$

$$\underline{\mathbf{S}}_2 = \left[\mathbf{I}_{L \times L} - \underline{\mathbf{S}}_{12}^2 (\underline{\mathbf{S}}_{11}^{-1} \underline{\mathbf{S}}_{22}^{-1}) \right] \underline{\mathbf{S}}_{22}. \quad (5.115b)$$

From (5.100), the squared interchannel coherence function of the f^{th} frequency bin may be expressed in terms of the spectra of input signals as

$$|\gamma(f)|^2 = \frac{|S_{12}(f)|^2}{S_{11}(f)S_{22}(f)}, \quad (5.116)$$

for $f = 0, 1, \dots, L-1$ hence giving the $L \times L$ diagonal squared coherence matrix

$$|\Gamma|^2 = \text{diag}\{|\gamma(0)|^2 |\gamma(1)|^2 \dots |\gamma(L-1)|^2\}. \quad (5.117)$$

The diagonal matrices $\underline{\mathbf{S}}_1^{-1}$ and $\underline{\mathbf{S}}_2^{-1}$ of (5.114) can now be expressed in terms of (5.117)

as

$$\mathbf{S}_1^{-1} = [\mathbf{I}_{L \times L} - |\Gamma|^2]^{-1} \underline{\mathbf{S}}_{11}^{-1}, \quad (5.118a)$$

$$\mathbf{S}_2^{-1} = [\mathbf{I}_{L \times L} - |\Gamma|^2]^{-1} \underline{\mathbf{S}}_{22}^{-1}, \quad (5.118b)$$

from which (5.113) can now be simplified as

$$\begin{aligned} \text{tr}\{\underline{\mathbf{S}}^{-1}\} &= \text{tr}\{\mathbf{R}_{xx}^{-1}\} \\ &= \sum_{l=0}^{L-1} [1 - |\gamma(l)|^2]^{-1} [S_{11}^{-1}(l) + S_{22}^{-1}(l)]. \end{aligned} \quad (5.119)$$

Substituting (5.111) and (5.119) into (5.109), the relationship between interchannel coherence and E-norm condition number of \mathbf{R}_{xx} can finally be expressed [31] as

$$\chi_E^2[\mathbf{R}_{xx}^{1/2}] = \frac{1}{4L^2} \left[\sum_{l=0}^{L-1} [S_{11}(l) + S_{22}(l)] \right] \left[\sum_{l=0}^{L-1} [1 - |\gamma(l)|^2]^{-1} [S_{11}^{-1}(l) + S_{22}^{-1}(l)] \right]. \quad (5.120)$$

Note that the computation of $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ is tractable since $\underline{\mathbf{S}}_{11}$, $\underline{\mathbf{S}}_{22}$, and Γ are diagonal matrices. More importantly, it is now evident from (5.120) that $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ increases with the squared interchannel coherence function hence degrading the condition of \mathbf{R}_{xx} . Figure 5.4 shows how $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ varies with the mean of $|\gamma(f)|^2$ across frequency bins $0 \leq f \leq L - 1$ for an example case of $L = 1024$ with stereophonic inputs generated using a zero mean WGN source. Using

$$\eta(n) = 10 \log_{10} \frac{\|\mathbf{h}(n) - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}(n)\|_2^2} \text{ dB} \quad (5.121)$$

as the normalized misalignment, Fig. 5.5 shows how the steady-state normalized misalignment of the two-channel frequency-domain adaptive algorithm [11], shown in Table 5.4 of Section 5.8.7, degrades with increasing $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ using an SNR of 40 dB. Hence, it can be observed that as the mean of $|\gamma(f)|^2 \rightarrow 0$, the E-norm condition number $\chi_E^2[\mathbf{R}_{xx}^{1/2}] \rightarrow 1$ and a good misalignment performance is expected. In addition, as the mean of $|\gamma(f)|^2 \rightarrow 1$, $\chi_E^2[\mathbf{R}_{xx}^{1/2}] \rightarrow \infty$ such that steady-state normalized misalignment performance degrades significantly. Consequently, for realistic SAEC applications with squared interchannel coherence in the range of approximately 0.95 to 0.97, poor misalignment performance is

5.4 Effect of interchannel coherence on the conditioning of \mathbf{R}_{xx} for SAEC 175

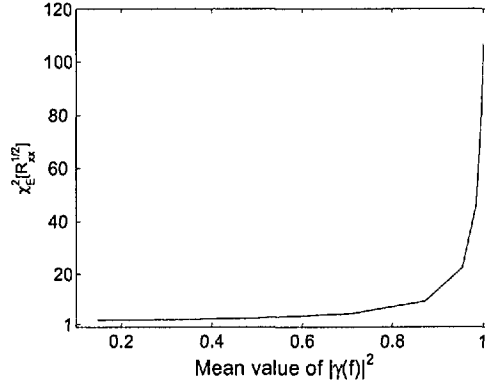


Figure 5.4: Variation of E-norm condition number $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$ with the mean of $|\gamma(f)|^2$ across frequency bins $0 \leq f \leq L - 1$.

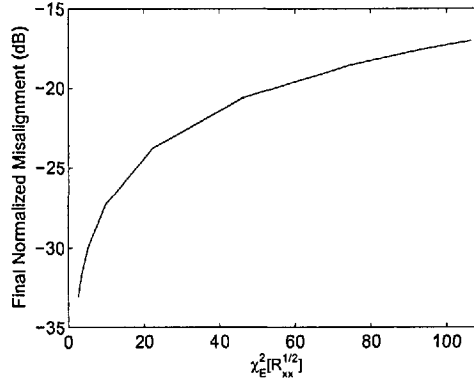


Figure 5.5: Variation of steady-state normalized misalignment with E-norm condition number $\chi_E^2[\mathbf{R}_{xx}^{1/2}]$.

expected unless the interchannel coherence is reduced.

5.4.5 Application to two-channel frequency-domain adaptive algorithm

This section examines how the formulation above can be applied to a two-channel frequency-domain adaptive algorithm to estimate its misalignment, hence verifying the relationship given in (5.120). It has been shown [11] in Section 5.3.4 that the steady-state normalized misalignment after convergence can be approximated by

$$\eta(n) \approx 10 \log_{10} \left[\frac{(1 - \lambda)}{2} \frac{\sigma_w^2}{\|\mathbf{h}\|_2^2} \text{tr}\{\mathbf{S}^{-1}\} \right] \text{ dB} , \quad (5.122)$$

from which using (5.113) and (5.120) for this two-channel case, can be further expressed as

$$\eta(n) \approx 10 \log_{10} \left[\frac{(1-\lambda)}{2} \frac{\sigma_w^2}{\|\mathbf{h}\|_2^2} \text{tr}\{\mathbf{R}_{\mathbf{xx}}^{-1}\} \right] \quad (5.123a)$$

$$\approx 10 \log_{10} \left[\frac{(1-\lambda)4L}{2} \frac{\sigma_w^2}{\|\mathbf{h}\|_2^2 \sigma_x^2} \chi_E^2[\mathbf{R}_{\mathbf{xx}}^{1/2}] \right], \quad (5.123b)$$

where σ_w^2 and $\sigma_x^2 = \sigma_{x_1}^2 + \sigma_{x_2}^2$ are the noise and input signal variances respectively. Hence, the misalignment is a function of the forgetting factor $0 \ll \lambda < 1$, signal-to-noise ratio (SNR) and the condition number $\chi_E^2[\mathbf{R}_{\mathbf{xx}}^{1/2}]$.

The two-channel frequency-domain adaptive algorithm [11], as shown in Table 5.4 of Section 5.8.7, has been shown to achieve good convergence performance for SAEC. It should be noted that for practicality, $\eta(n)$ can be computed using (5.123a). However, in order to verify the validity of (5.120), $\chi_E^2[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ is first computed using (5.120) such that elements $|\gamma(l)|^2$ are estimated using the interchannel coherence estimate $|\widehat{\Gamma}(m)|^2$ given in Table 5.4. The theoretical steady-state normalized misalignment is then computed by employing (5.123b). The validity of the established link is verified through simulations in Section 5.6. It is also evident from (5.120) and (5.123b) that high interchannel coherence will degrade the conditioning of $\mathbf{R}_{\mathbf{xx}}$ hence reducing the performance of the adaptive algorithm in terms of its steady-state misalignment as shown in Fig. 5.4 and 5.5. In addition, the formulation presented in this section is more general than [114] since $\underline{\mathbf{S}}_{jj}$ has not been assumed to be constant across frequency. This is feasible especially for speech signals where the spectra is not constant across frequency as will be shown through simulation examples presented in Section 5.6.

5.5 Frequency-domain adaptive filtering employing XM tap selection

In this section, frequency-domain algorithms employing exclusive-maximum (XM) tap selection will be developed. Drawing upon the link between interchannel coherence and condition number of $\mathbf{R}_{\mathbf{xx}}$ as described in Section 5.4, this section examines how the XM tap selection can reduce the interchannel coherence which in turn improves the conditioning

of \mathbf{R}_{xx} giving good convergence performance. Two cases of tap selection, which can be achieved either by subselecting the time-domain or the frequency-domain tap-input vector, will be considered. Their impact to the convergence performance of frequency-domain algorithms will also be examined in the context of the M-ratio measure \mathcal{M} . The XM tap selection is then extended to the frequency-domain FLMS algorithm by first considering a 50% overlapping factor between successive input blocks and an arbitrary overlapping factor controlled by the variable $\alpha > 1$.

5.5.1 Effect of XM tap selection on interchannel coherence and condition number of \mathbf{R}_{xx}

As noted from Fig. 4.5, exclusive tap selection can improve the conditioning of \mathbf{R}_{xx} . In this section, the effect of exclusive tap selection² on the interchannel coherence and conditioning of \mathbf{R}_{xx} is analyzed mathematically by exploiting frequency-domain quantities and the E-norm condition number as depicted in Section 5.4. By virtue of the exclusivity constraint imposed on the two-channel tap-input vectors, it can be shown for tap selection control matrices $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$ that

$$\begin{aligned} \mathbf{Q}_1(n) \odot \mathbf{Q}_2(n) &= \mathbf{Q}_1(n)\mathbf{Q}_2(n) \\ &= \mathbf{0}_{L \times L}, \end{aligned} \tag{5.124}$$

where \odot is defined as the element-by-element (Schür) product and $\mathbf{0}_{L \times L}$ is the $L \times L$ null matrix. The cross-correlation function $r_{12}(l)$ between $x_1(n)$ and $x_2(n)$ is defined as

$$r_{12}(l) = E\{x_1(n)x_2(n-l)\}. \tag{5.125}$$

²The exclusive tap selection discussed in this subsection is not limited to XM tap selection. Since the aim is to analyze only the decorrelation effects brought about by exclusive tap selection, an arbitrary exclusive selection such as selecting odd coefficient indices in channel 1 and even coefficient indices in channel 2 is considered.

With the j^{th} channel subselected tap-input vector given as $\tilde{\mathbf{x}}_j(n) = \mathbf{Q}_j(n)\mathbf{x}_j(n)$, the effect of the exclusivity constraint on the cross-correlation function (at zero-lag) is given by

$$\begin{aligned}
 r_{12}(0) &= r_{21}(0) \\
 &= E\{\mathbf{Q}_1(n)\mathbf{x}_1(n)\mathbf{Q}_2(n)\mathbf{x}_2(n)\} \\
 &= E\{\tilde{\mathbf{x}}_1(n)\tilde{\mathbf{x}}_2(n)\} \\
 &= 0.
 \end{aligned} \tag{5.126}$$

In addition, the cross-correlations $r_{12}(l)$ and $r_{21}(l)$ are “sparsified” by $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$ and as a consequence, with reference to (5.100), $|S_{12}(f)|^2$ and hence the squared-coherence $|\gamma(f)|^2$ defined in (5.116) are reduced accordingly. This reduction in interchannel coherence due to the exclusive tap selection can be observed from experimental results as presented in Fig. 4.4 (c). Additionally, from (5.120), a reduction in interchannel coherence reduces the E-norm condition number $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ and with the maximization of \mathcal{M} as described in Section 4.3.1, improved convergence performance of XM-based algorithms [30] is expected.

5.5.2 Selection in frequency-domain vs selection in time-domain

One of the main concerns in developing frequency-domain adaptive algorithms employing tap selection is to consider whether subselection should be performed in the frequency- or time-domain. These two options are analyzed and considered. For simplicity, a single channel case is initially considered where the MMax tap selection criterion [21] is imposed on the tap-input vector³. With reference to the FLMS algorithm as discussed in Section 5.3.1 and tap-input vector $\mathbf{X}(m)$ as defined in (5.3), tap selection can be achieved by first considering the case of subselecting a frequency transformed tap-input vector, i.e.,

$$\begin{aligned}
 \tilde{\mathbf{X}}_f(m) &= \mathbf{Q}(m)[\mathbf{F}_{2L \times 2L}\mathbf{X}(m)] \\
 &= \mathbf{Q}(m)\underline{\mathbf{X}}(m),
 \end{aligned} \tag{5.127}$$

where the subscript f in $\tilde{\mathbf{X}}_f(m)$ denotes subselection in the frequency-domain and tap-input vector $\underline{\mathbf{X}}(m)$ is defined in (5.5). The elements of the diagonal MMax tap selection matrix

³Using the MMax tap selection for the single channel case here allows one to simplify this discussion and employ the measure \mathcal{M} to quantify the effect of subselection. The impact of using the single channel MMax tap selection on the XM tap selection for stereo case will be discussed at the end of this subsection.

$\mathbf{Q}(m)$ in this case is given by

$$q_l(n) = \begin{cases} 1, & |x_l(m)| \in \{M \text{ maxima of } |\underline{\mathbf{X}}(m)|\}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.128)$$

for $l = 0, 1, \dots, 2L - 1$ where $|x_l(m)|$ is the l^{th} element of $|\underline{\mathbf{X}}(m)|$ given that

$$|\underline{\mathbf{X}}(m)| = [|x_0(m)| |x_1(m)| \dots |x_{2L-1}(m)|]^T. \quad (5.129)$$

Consider an alternative case where tap selection is achieved instead by subselecting the time-domain tap-input vector $\mathbf{X}(m)$ *before* being transformed to the frequency-domain. In this case the subselected tap-input vector can be expressed as

$$\begin{aligned} \tilde{\underline{\mathbf{X}}}_t(m) &= \mathbf{F}_{2L \times 2L} [\mathbf{Q}(m) \mathbf{X}(m)] \\ &= \mathbf{F}_{2L \times 2L} \tilde{\underline{\mathbf{X}}}(m), \end{aligned} \quad (5.130)$$

where the subscript t in $\tilde{\underline{\mathbf{X}}}_t(m)$ denotes subselection in the time-domain. Elements of the diagonal tap selection matrix $\mathbf{Q}(m)$ are now given, for $l = 0, 1, \dots, 2L - 1$, by

$$q_l(n) = \begin{cases} 1, & |x(mL - L + l)| \in \{M \text{ maxima of } |\mathbf{X}(m)|\}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.131)$$

where the time-domain tap-input vector $\mathbf{X}(m)$ is defined by (5.3) and $|x(mL - L + l)|$ is the l^{th} element in $|\mathbf{X}(m)|$ given that

$$|\mathbf{X}(m)| = [|x(mL - L)| |x(mL - L + 1)| \dots |x(mL + L - 1)|]^T. \quad (5.132)$$

As discussed in Section 4.3.1, the use of \mathcal{M} defined in (4.18) allows one to analyze the effect of tap selection on the rate of convergence. With the subselected tap-input vectors expressed in (5.127) and (5.130), the effect of tap selection for $\tilde{\underline{\mathbf{X}}}_f(m)$ and $\tilde{\underline{\mathbf{X}}}_t(m)$ on their

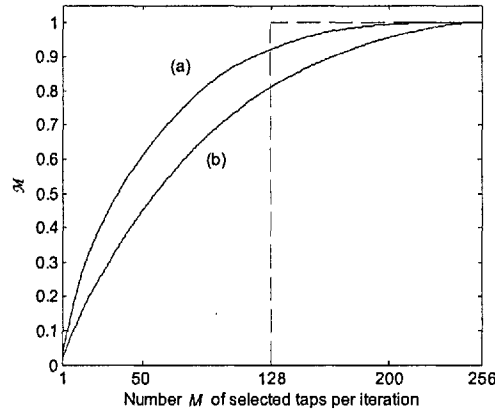


Figure 5.6: Effect of tap selection on (a) \mathcal{M}_t and (b) \mathcal{M}_f for an adaptive filter length of $L = 256$.

respective \mathcal{M} can be compared by computing $\mathcal{M}_f(m)$ and $\mathcal{M}_t(m)$ respectively as

$$\mathcal{M}_f(m) = \frac{\|\mathbf{F}_{2L \times 2L}^{-1} \tilde{\mathbb{X}}_f(m)\|_2^2}{\|\mathbb{X}(m)\|_2^2}, \quad (5.133a)$$

$$\mathcal{M}_t(m) = \frac{\|\mathbf{F}_{2L \times 2L}^{-1} \tilde{\mathbb{X}}_t(m)\|_2^2}{\|\mathbb{X}(m)\|_2^2}. \quad (5.133b)$$

Figure 5.6 shows the effect of tap selection on M-ratios (a) \mathcal{M}_t and (b) \mathcal{M}_f . In this illustrative example, an adaptive filter of length $L = 256$ is used with a zero mean unit variance WGN input sequence. It can be observed that for all cases of tap selection size M , $\mathcal{M}_f < \mathcal{M}_t$. More importantly, for $M = 0.5L = 128$, there is a significant reduction in \mathcal{M}_f compared to \mathcal{M}_t . Since, as explained in Section 4.3.1, the rate of convergence is a monotonic increasing function of \mathcal{M} , it is proposed that

- *the degradation in convergence performance due to tap selection for a frequency-domain algorithm employing MMax tap selection can be reduced by subselecting the tap-input vector in the time-domain before taking its Fourier transform for adaptation.*

Although the analysis described here concerns single channel MMax tap selection, extension to the two-channel XM tap selection is direct and straightforward since the XM tap selection jointly maximizes \mathcal{M} in both channels. Consequently, for the j^{th} channel, subselection of the time-domain tap-input vector $\mathbb{X}_j(m)$ will be employed instead of $\tilde{\mathbb{X}}_j(m)$ for the development of XM-based frequency-domain algorithms for SAEC such as shown

below.

5.5.3 Tap selection with 50% overlapping factor

With reference to Fig. 5.2 the j^{th} channel tap-input vector for FLMS is described by (5.3) for $j = 1, 2$ and is reproduced here for convenience

$$\mathbb{X}_j(m) = [x_j(mL-L) x_j(mL-L+1) \dots x_j(mL-1) x_j(mL) x_j(mL+1) \dots x_j(mL+L-1)]^T. \quad (5.134)$$

This $2L \times 1$ vector can be decomposed into two sub-vectors giving

$$\begin{aligned} \mathbb{X}_j(m) &= \left[\underbrace{x_j(mL-L) \dots x_j(mL-1)}_{\mathbb{X}_{b,j}^T(m-1)} \underbrace{x_j(mL) \dots x_j(mL+L-1)}_{\mathbb{X}_{b,j}^T(m)} \right]^T \\ &= [\mathbb{X}_{b,j}^T(m-1) \mathbb{X}_{b,j}^T(m)]^T, \end{aligned} \quad (5.135)$$

where the $L \times 1$ sub-vector $\mathbb{X}_{b,j}(m)$ is defined by

$$\mathbb{X}_{b,j}(m) = [x_j(mL) x_j(mL+1) \dots x_j(mL+L-1)]^T, \quad (5.136)$$

such that the first subscript b denotes a sub-vector of $\mathbb{X}_j(m)$. Defining the $L \times 1$ magnitude difference vector computed for each frame

$$\mathbf{p}(m) = |\mathbb{X}_{b,1}(m)| - |\mathbb{X}_{b,2}(m)|, \quad (5.137)$$

the subselected tap-input vector is then given by

$$\tilde{\mathbb{X}}_{b,j}(m) = \mathbf{Q}_j(m) \mathbb{X}_{b,j}(m), \quad (5.138)$$

where the j^{th} channel diagonal $L \times L$ diagonal XM tap selection matrix is defined as

$$\mathbf{Q}_j(m) = \text{diag}\{\mathbf{q}_j(m)\}. \quad (5.139)$$

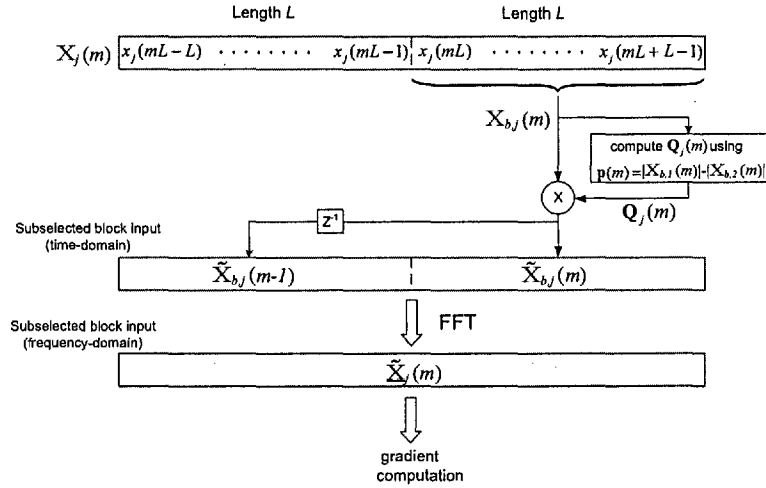


Figure 5.7: Schematic of subselected tap-input vector and its frequency-domain quantity using 50% overlapping factor.

As shown in (4.42a) and (4.42b), element u of \mathbf{q}_1 and element v of \mathbf{q}_2 are defined for $u, v = 0, 1, \dots, L-1$ and $M = 0.5L$ where

$$q_{1,u} = \begin{cases} 1, & p_u \in \{M \text{ maxima of } \mathbf{p}(m)\}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.140a)$$

$$q_{2,v} = \begin{cases} 1, & p_v \in \{M \text{ minima of } \mathbf{p}(m)\}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.140b)$$

For practical implementation, the j^{th} channel sub-vector $\mathbb{X}_{b,j}(m-1)$ is a delayed version of $\mathbb{X}_{b,j}(m)$ and in a similar manner, $\tilde{\mathbb{X}}_{b,j}(m-1)$ is obtained from $\tilde{\mathbb{X}}_{b,j}(m)$ using a delay. The $2L \times 1$ subselected tap-input vector for the m^{th} frame can then be expressed as

$$\tilde{\mathbb{X}}_j(m) = \left[\tilde{\mathbb{X}}_{b,j}^T(m-1) \tilde{\mathbb{X}}_{b,j}^T(m) \right]^T, \quad (5.141)$$

from which the corresponding j^{th} channel frequency-domain subselected tap-input vector is obtained using

$$\tilde{\mathbb{X}}_j(m) = \mathbf{F}_{2L \times 2L} \tilde{\mathbb{X}}_j(m). \quad (5.142)$$

Similar to the FLMS algorithm, $\tilde{\mathbb{X}}_j(m)$ can then be employed for gradient computation. To illustrate above, Fig. 5.7 shows a schematic of how tap selection can be achieved using a 50% overlapping factor. The proposed FLMS algorithm incorporating XM tap se-

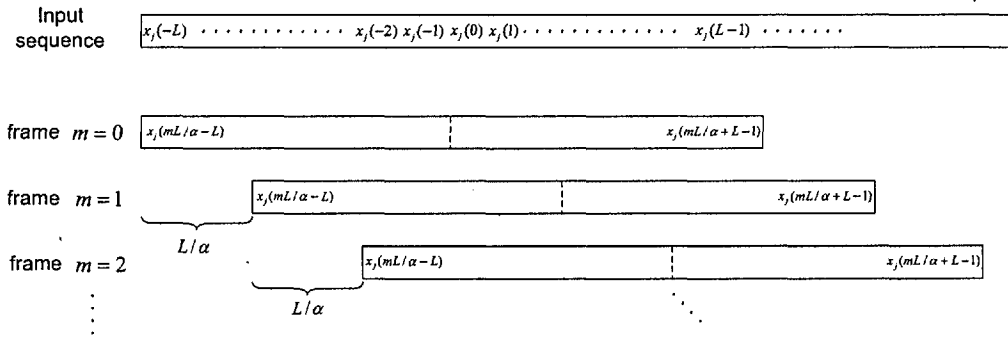


Figure 5.8: Input sequence partitioning for the j^{th} channel FLMS algorithm with arbitrary overlapping factor controlled by α .

lection (XM-FLMS) [32] is depicted in Table 5.5 of Section 5.8.7. Similar to time-domain implementations as shown in Chapter 4, the use of XM tap selection is proposed in combination with the non-linear (NL) preprocessor to improve the convergence rate of FLMS in combination with NL-preprocessor (NL-FLMS) and as such, this proposed algorithm is denoted as XMNL-FLMS. Performance comparison between XMNL-FLMS and NL-FLMS will be presented through simulations in Section 5.6.

5.5.4 Tap selection with arbitrary overlapping factor

Instead of a 50% overlap between successive tap-input vectors as shown in Fig. 5.2, the XM tap selection can be extended to the FLMS algorithm using an arbitrary overlapping factor similar to the GMDF α algorithm [110]. The single channel GMDF α algorithm incorporates the MDF [23] structure as described in Section 5.3.2 to reduce the delay inherent in frequency-domain approaches by partitioning the adaptive filter into \mathcal{K} blocks. In addition, GMDF α achieves fast convergence by employing an arbitrary overlapping factor between successive tap-input frames controlled by an overlapping factor $\alpha \geq 1$. With $\alpha > 1$, the filter coefficients are updated more frequently and for a large step-size, the GMDF α algorithm has been shown to achieve a faster rate of convergence compared to the MDF algorithm [110]. Figure 5.8 shows how the input sequence is partitioned using an arbitrary overlapping factor α . For $\alpha = 1$, a 50% overlap between successive input blocks is achieved as shown in Fig. 5.2. Since the aim of this work is to introduce tap selection for any arbitrary $\alpha \geq 1$, the case of a single adaptive filter block $\mathcal{K} = 1$ for each channel will be considered in this section and as a result the proposed algorithm is denoted

as XM-FLMS α [32].

To incorporate the XM tap selection to a tap-input vector obtained from an arbitrary overlapping factor, the j^{th} channel input signal $x_j(n)$, $j = 1, 2$, is partitioned into overlapping sections each having size $2L$. Direct deployment of the $\mathbf{Q}_j(m)$ diagonal matrix, such as expressed by (5.138) is now inappropriate since $\mathbf{Q}_j(m)$ is of dimension $L \times L$. However, subselection at each block iteration can be incorporated by first denoting the $2L \times 1$ tap-input sequence for the j^{th} channel as

$$\mathbb{X}_{\alpha,j}(m) = [x_j(mL/\alpha - L) \ x_j(mL/\alpha - L + 1) \ \dots \ x_j(mL/\alpha + L - 1)]^T, \quad (5.143)$$

where $\alpha \geq 1$ controls the overlapping between successive input frames and the subscript α in $\mathbb{X}_{\alpha,j}$ shows the dependency of input frames on α . With this notation and for $\alpha = 1$, it can be observed that $\mathbb{X}_{\alpha,j}(m) = \mathbb{X}_j(m)$ where $\mathbb{X}_j(m)$ has been defined in (5.134). Similar to (5.137), the $2L \times 1$ magnitude difference vector can then be defined as

$$\mathbf{p}(m) = |\mathbb{X}_{\alpha,1}(m)| - |\mathbb{X}_{\alpha,2}(m)|, \quad (5.144)$$

from which the XM tap selection criterion for this arbitrary overlapping case is defined for element u of \mathbf{q}_1 and element v of \mathbf{q}_2 for $u, v = 0, 1, \dots, 2L - 1$ where

$$q_{1,u} = \begin{cases} 1, & p_u \in \{M \text{ maxima of } \mathbf{p}(m)\}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.145a)$$

$$q_{2,v} = \begin{cases} 1, & p_v \in \{M \text{ minima of } \mathbf{p}(m)\}, \\ 0, & \text{otherwise.} \end{cases} \quad (5.145b)$$

Note that the vector $\mathbf{p}(m)$ is now of length $L' = 2L$. Since half the number of taps corresponding to the maximum values of $\mathbf{p}(m)$ in the first channel have to be selected, for this XM-FLMS α case, $M = 0.5L' = L$ will be selected in (5.145a) and (5.145b). Consequently, the j^{th} channel diagonal tap selection matrix $\mathbf{Q}_j(m)$ is of dimension $2L \times 2L$ which results in a subselected tap-input vector

$$\tilde{\mathbb{X}}_{\alpha,j} = \mathbf{Q}_j(m)\mathbb{X}_{\alpha,j}(m). \quad (5.146)$$

In a similar manner to the XM-FLMS algorithm, the frequency-domain tap-input vector for XM-FLMS α is then computed using

$$\tilde{\mathbf{X}}_{\alpha,j} = \mathbf{F}_{2L \times 2L} \tilde{\mathbf{X}}_{\alpha,j}, \quad (5.147)$$

which is then used for gradient computation, similar to that shown in (5.16).

Comparing (5.146) and (5.138), the XM-FLMS α requires an additional L addition/subtraction operations since the 50% block delay cannot be applied here. The proposed XM-FLMS α is depicted in Table 5.6 of Section 5.8.7. Similar to the FLMS algorithm, the use of XM tap selection is proposed in combination with the non-linear (NL) preprocessor [52] and hence will be denoted as XMNL-FLMS α .

5.6 Simulation results

5.6.1 Verification of (5.120)

The link between interchannel coherence and the E-norm condition number $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ given by (5.120) is verified using the two-channel frequency-domain adaptive filtering algorithm [11] given in Table 5.4 of Section 5.8.7. In these simulations, the lengths of both the adaptive filters are $L = 1024$ with $\lambda = [1 - 1/(3L)]^L$ and $\mu = 2$. The stereophonic impulse responses of both the transmission and receiving rooms are recorded at 16 kHz sampling rate and are of length 4096. To neglect any misalignment effects due to under-modelling, the impulse responses of the receiving room are truncated to 1024. Figure 5.9 shows the normalized misalignment plots for a zero mean unit variance WGN source sequence where input vectors $\mathbf{X}_1(m)$ and $\mathbf{X}_2(m)$ are generated by convolving this source with the impulse responses of the transmission room. With reference to Fig. 4.1, an uncorrelated zero mean WGN sequence $w(n)$ is added to the received signal to achieve SNRs of 25 and 35 dB.

The interchannel coherence is varied using a non-linearity control factor β [52] given by (4.46a) and (4.46b). Theoretical steady-state normalized misalignments, shown as straight horizontal lines, are computed and are averaged across block iterations using input signals $\mathbf{X}'_1(m)$ and $\mathbf{X}'_2(m)$. Although the non-linearity control is used to vary the

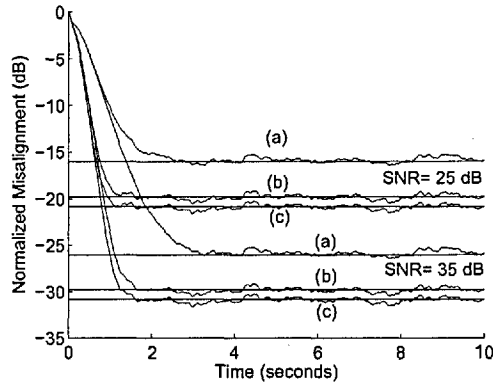


Figure 5.9: Normalized misalignment for WGN input with mean interchannel coherences of (a) 0.85, (b) 0.60 and (c) 0.53.

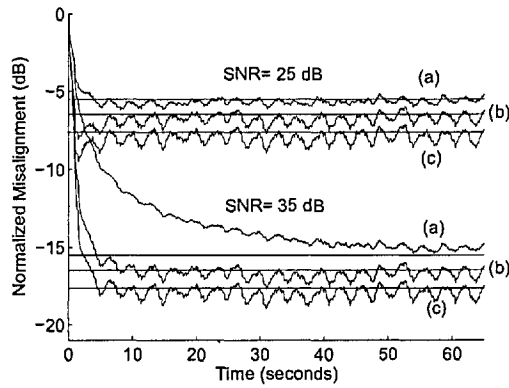


Figure 5.10: Normalized misalignment for speech input with mean interchannel coherences of (a) 0.85, (b) 0.60 and (c) 0.53.

interchannel coherence, the analysis presented does not make any assumptions about the methods of achieving this variation⁴. In order to verify (5.120), the normalized misalignment is computed using (5.120) and (5.123b). Due to the variation of β , the measured mean interchannel coherences across frequency between $\mathbb{X}'_1(m)$ and $\mathbb{X}'_2(m)$ are (a) 0.85, (b) 0.60 and (c) 0.53. It can be observed that as $\mathbb{X}'_1(m)$ and $\mathbb{X}'_2(m)$ become more uncorrelated, the steady-state normalized misalignment reduces gracefully as expected. The theoretical normalized steady-state misalignments computed using (5.120) are also consistent with the experimental results hence verifying the relationship between interchannel coherence and condition number $\chi_E[\mathbf{R}_{xx}]^{1/2}$.

⁴Additional results for various interchannel coherences achieved by adding WGN to $x_2(m)$ is shown in Section 5.8.6

Figure 5.10 shows normalized misalignment plots using the same experimental setup as above but with speech input sequence from a male speaker. As before, the variation of interchannel coherence is controlled using the non-linearity constant β such that the measured mean interchannel coherences across frequency are the same as before. The mean theoretical normalized steady-state misalignments across time iterations are plotted as straight horizontal lines. It can be seen that the normalized misalignment performance degrades with increasing interchannel coherence as expected and the theoretical normalized steady-state misalignment computed using $\chi_E^2[\mathbf{R}_{\mathbf{x}\mathbf{x}}^{1/2}]$ is consistent with the experimental results hence verifying (5.120).

5.6.2 Experimental setup for FLMS and FLMS α based simulations

Simulation results are presented to illustrate the convergence performances of XM-based frequency-domain adaptive algorithms. For all simulations shown below, impulse responses $\mathbf{g}_1(n)$, $\mathbf{g}_2(n)$, $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$ are generated using the method of images [72]. Two microphones are placed 1 m apart in the centre of both the transmission and receiving rooms each of dimension $3 \times 4 \times 5$ m. The source is then positioned 1 m away from each microphone in the transmission room. Tap-input vectors $\mathbb{X}'_1(m)$ and $\mathbb{X}'_2(m)$ are obtained by convolving the source with two impulse responses $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ and then applying the non-linear (NL) preprocessor defined in (4.46a) and (4.46b) with a non-linear control factor of $\beta = 0.5$. An uncorrelated zero mean WGN sequence $w(n)$ is added to achieve an SNR of 25 dB. For clarity, the normalized misalignment of only one channel is plotted in each experiment.

5.6.3 FLMS with 50% overlapping-factor simulations

The performance of NL-FLMS is compared with that of the XMNL-FLMS algorithm as shown in Fig. 5.11 (a) and (b) respectively using a zero mean unit variance WGN source sequence. In this simulation, the lengths of the adaptive filters are $L = 256$ while the lengths of the transmission and receiving rooms are $L_T = 800$ and $L_R = 800$ respectively with a reverberation time of $T_{60} = 100$ ms. A sampling frequency of $f_s = 8$ kHz is used while a non-linearity control factor [52] of $\beta = 0.5$ is used. The step-size of the NL-FLMS algorithm is $\mu_{\text{NL-FLMS}} = 1$ while for the XMNL-FLMS, $\mu_{\text{XMNL-FLMS}} = 0.4$ is

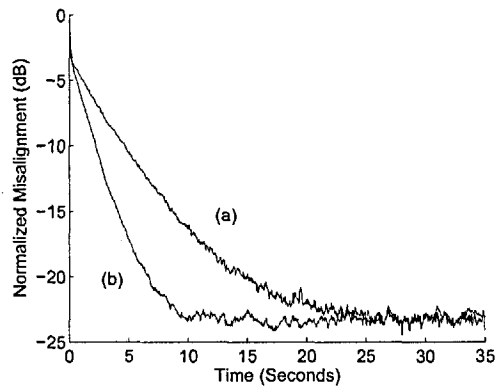


Figure 5.11: Normalized misalignment using WGN input sequence for (a) NL-FLMS and (b) XMNL-FLMS [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-FLMS}} = 1$, $\mu_{\text{XMNL-FLMS}} = 0.4$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

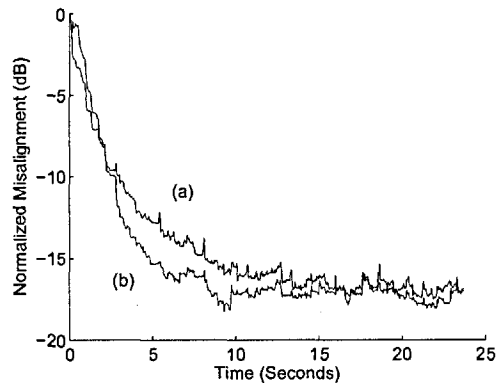


Figure 5.12: Normalized misalignment using speech input sequence for (a) NL-FLMS and (b) XMNL-FLMS [$L_T = L_R = 800$, $L = 256$, $\mu_{\text{NL-FLMS}} = 1$, $\mu_{\text{XMNL-FLMS}} = 0.43$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

used to achieve the same steady-state normalized misalignment. From Fig. 5.11, it can be seen that due to the XM tap selection, XMNL-FLMS outperforms the NL-FLMS by approximately 7 to 8 dB normalized misalignment during convergence. Alternatively, the NL-FLMS algorithm requires an additional 12 s before achieving the same steady-state misalignment as the XMNL-FLMS algorithm.

Figure 5.12 shows simulation results for (a) NL-FLMS and (b) XMNL-FLMS using the same experimental setup as above but with speech input sequence from a male talker. The SNR was computed using the whole utterance of the speech sequence. The step-size of the XMNL-FLMS algorithm is $\mu_{\text{XMNL-FLMS}} = 0.43$ in order to achieve the same steady-

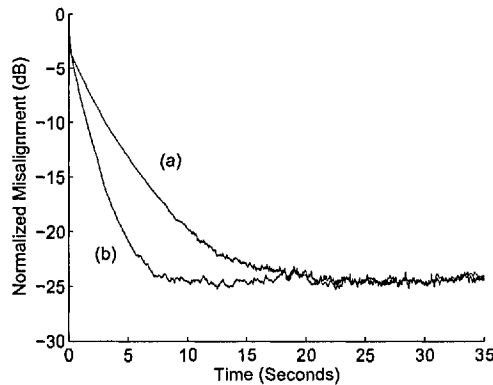


Figure 5.13: Normalized misalignment using WGN input sequence for (a) NL-FLMS α and (b) XMNL-FLMS α [$L_T = L_R = 800$, $L = 256$, $\alpha = 4$, $\mu_{\text{NL-FLMS}\alpha} = 1$, $\mu_{\text{XMNL-FLMS}\alpha} = 0.65$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

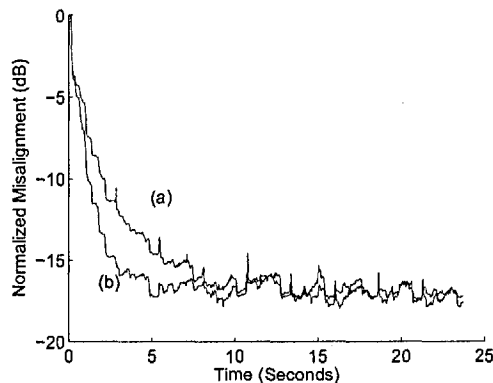


Figure 5.14: Normalized misalignment using speech input sequence for (a) NL-FLMS α and (b) XMNL-FLMS α [$L_T = L_R = 800$, $L = 256$, $\alpha = 4$, $\mu_{\text{NL-FLMS}\alpha} = 1$, $\mu_{\text{XMNL-FLMS}\alpha} = 0.65$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms, SNR = 25 dB].

state misalignment as the NL-FLMS algorithm where $\mu_{\text{NL-FLMS}} = 1$ as before. For this speech input example, it can be observed that the XMNL-FLMS algorithm outperforms NL-FLMS by approximately 3 to 4 dB of normalized misalignment during convergence. The NL-FLMS algorithm requires an additional 6 s before reaching the same steady-state misalignment as the XMNL-FLMS algorithm.

5.6.4 FLMS with arbitrary overlapping-factor simulations

The performance of XMNL-FLMS α is illustrated by comparing its convergence to that of NL-FLMS α as shown in Fig. 5.13 using a zero mean unit variance WGN source sequence.

As before, the parameters for this experiment are $L = 256$, $L_T = 800 = L_R = 800$, $\beta = 0.5$, $f_s = 8$ kHz, $T_{60} = 100$ ms and $\text{SNR} = 25$ dB. Step-sizes $\mu_{\text{NL-FLMS}\alpha} = 1$ and $\mu_{\text{XMNL-FLMS}\alpha} = 0.65$ are used for NL-FLMS α and XMNL-FLMS α respectively such that they achieve the same steady-state normalized misalignment. In this simulation, an arbitrarily chosen overlapping control factor of $\alpha = 4$ is used. It can be seen from Fig. 5.13 that the XMNL-FLMS α algorithm outperforms NL-FLMS α by approximately 8 dB in normalized misalignment during convergence. Alternatively, the NL-FLMS algorithm requires an additional 15 s before reaching the same steady-state normalized misalignment.

Figure 5.14 compares the misalignment performances of (a) NL-FLMS α and (b) XMNL-FLMS α using the same experimental setup as above but with a speech input sequence from a male talker. The SNR was computed using the whole utterance of the speech sequence. As before, due to the reduction in interchannel coherence brought about by XM tap selection, the XMNL-FLMS α algorithm outperforms NL-FLMS α by approximately 3 dB. The NL-FLMS α algorithm requires an additional 6 s before reaching the same steady-state misalignment as XMNL-FLMS α . Comparing Figs. 5.12 and 5.14, the rate of convergence for the FLMS α -based algorithms is higher than for FLMS-based algorithms since, similar to the GMDF α algorithm [110], the adaptive filters are being updated more frequently for an overlapping control factor of $\alpha = 4$.

5.7 Conclusions

In this chapter, the fast-LMS (FLMS) [22] and MDF [23] algorithms have been reviewed. The derivation of these algorithms using a frequency-domain cost function [11] has also been reviewed. The link between interchannel coherence and conditioning of the two-channel input correlation matrix is established [31] by exploiting frequency-domain quantities and the E-norm condition number [111]. It has been shown how the high interchannel coherence degrades the conditioning of \mathbf{R}_{xx} and using this relationship, it has been explained how the XM tap selection reduces the interchannel coherence hence improving the conditioning of \mathbf{R}_{xx} . In order to reduce the degradation of \mathcal{M} , tap selection is employed on the time-domain tap-input vector before taking its Fourier transform. Using this tap selection approach, the XM tap selection is then extended to frequency-domain adaptive algorithms employing both the 50% and an arbitrary overlapping factor controlled by α .

Simulation results have been presented to verify the validity of the established relationship between interchannel coherence and $\chi_E[\mathbf{R}_{xx}^{1/2}]$. The convergence performances of the proposed XMNL-FLMS and XMNL-FLMS α algorithms showed improvement over NL-FLMS and NL-FLMS α for both WGN and speech input signals.

5.8 Appendix

5.8.1 Proof of matrix multiplication $(\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01}$

The matrix $(\mathbf{G}_{N \times 2N}^{01})^H$ can be simplified as

$$\begin{aligned}
 (\mathbf{G}_{N \times 2N}^{01})^H &= (\mathbf{F}_{N \times N} \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1})^H \\
 &= (\mathbf{F}_{2N \times 2N}^{-1})^H (\mathbf{W}_{N \times 2N}^{01})^H \mathbf{F}_{N \times N}^H \\
 &= [(\mathbf{F}_{2N \times 2N}^{-1})^*]^T \mathbf{W}_{2N \times N}^{01} [\mathbf{F}_{N \times N}^*]^T \\
 &= \left[\frac{1}{2N} \mathbf{F}_{2N \times 2N} \right]^T \mathbf{W}_{2N \times N}^{01} [N \mathbf{F}_{N \times N}^{-1}]^T \\
 &= 0.5 \times \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{F}_{N \times N}^{-1}
 \end{aligned} \tag{5.148}$$

where $\mathbf{W}_{2N \times N}^{01}$ is defined in (5.2a). Hence,

$$\begin{aligned}
 (\mathbf{G}_{N \times 2N}^{01})^H \mathbf{G}_{N \times 2N}^{01} &= 0.5 \times \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{W}_{N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1} \\
 &= 0.5 \times \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1} \\
 &= 0.5 \times \mathbf{G}_{2N \times 2N}^{01},
 \end{aligned} \tag{5.149}$$

where the windowing matrix $\mathbf{W}_{2N \times 2N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \end{bmatrix}_{2N \times 2N}$. □

5.8.2 Proof of equation (5.53)

Using (5.148),

$$\begin{aligned}
(\mathbf{G}_{N \times 2N}^{01})^H \underline{\mathbf{y}}(r) &= 0.5 \times \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{F}_{N \times N}^{-1} \underline{\mathbf{y}}(r) \\
&= 0.5 \times \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times N}^{01} \mathbf{F}_{N \times N}^{-1} \mathbf{F}_{N \times N} \mathbf{y}(r) \\
&= 0.5 \times \mathbf{F}_{2N \times 2N} \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{y}(r) \end{bmatrix}_{2N \times 1} \\
&= 0.5 \times \underline{\mathbf{y}}(r)
\end{aligned}$$

where (5.41) has been employed in the second step. \square

5.8.3 Proof of equation (5.77)

The relationship between $\underline{\mathbf{S}}(m)$ and $\underline{\mathbf{B}}(m)$ can be shown by postmultiplying $\mathbf{G}_{2L \times L}^{10}$, defined in (5.66), and premultiplying $(\mathbf{G}_{2L \times L}^{10})^H$ to (5.76) giving

$$\begin{aligned}
(\mathbf{G}_{2L \times L}^{10})^H \underline{\mathbf{B}}(m) \mathbf{G}_{2L \times L}^{10} &= (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} (\mathbf{G}_{2L \times L}^{10})^H \underline{\mathbf{D}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{D}}(r) \mathbf{G}_{2L \times L}^{10} \\
&= (1 - \lambda) \sum_{r=0}^m \lambda^{m-r} \underline{\mathbf{U}}^H(r) \mathbf{G}_{2N \times 2N}^{01} \underline{\mathbf{U}}(r) \\
&= \underline{\mathbf{S}}(m)
\end{aligned} \tag{5.150}$$

where (5.68) has been employed in the last step. \square

5.8.4 Proof of equation (5.78)

It can be shown that

$$\begin{aligned}
\mathbf{G}_{2L \times 2L}^{10} \mathbf{G}_{2L \times L}^{10} &= \mathbf{F}_{2L \times 2L} \mathbf{W}_{2L \times L}^{10} \mathbf{F}_{L \times L}^{-1} \\
&= \mathbf{G}_{2L \times L}^{10},
\end{aligned} \tag{5.151}$$

where matrices $\mathbf{G}_{2L \times 2L}^{10}$ and $\mathbf{W}_{2L \times 2L}^{10}$ are defined in (5.2g) and (5.2e) respectively. Employing the property $[\mathbf{AB}]^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1}$ [126] for arbitrary matrices \mathbf{A} and \mathbf{B} , the inverse

of $\underline{\mathbf{S}}^{-1}(m)$ in (5.77) can be expressed as

$$\underline{\mathbf{S}}^{-1}(m) = (\mathbf{G}_{2L \times L}^{10})^{-1} \underline{\mathbf{B}}^{-1}(m) [(\mathbf{G}_{2L \times L})^H]^{-1}. \quad (5.152)$$

Pre-multiplying and post-multiplying by $\mathbf{G}_{2L \times 2L}^{10} \mathbf{G}_{2L \times L}^{10}$ and $(\mathbf{G}_{2L \times 2L}^{10})^H$ respectively,

$$\mathbf{G}_{2L \times L}^{10} \underline{\mathbf{S}}^{-1}(m) (\mathbf{G}_{2L \times L}^{10})^H = \mathbf{G}_{2L \times 2L}^{10} \underline{\mathbf{B}}^{-1}(m).$$

□

5.8.5 Explicit link between $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ and $\chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ for uncorrelated $x_1(n)$ and $x_2(n)$

The independence of the E-norm condition number $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ on adaptive filter length L and its relationship with $\chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ can be shown for a simple case example by first assuming for zero mean WGN inputs $x_1(n)$ and $x_2(n)$ where $E\{x_1^2(n)\} = \sigma_{x_1}^2$ and $E\{x_2^2(n)\} = \sigma_{x_2}^2$ giving $L \times L$ matrices $\mathbf{R}_{11} = \text{diag}\{\sigma_{x_1}^2 \dots \sigma_{x_1}^2\}$ and $\mathbf{R}_{22} = \text{diag}\{\sigma_{x_2}^2 \dots \sigma_{x_2}^2\}$. For the case where $x_1(n)$ and $x_2(n)$ are perfectly uncorrelated, $\text{tr}\{\mathbf{R}_{\mathbf{xx}}\}$ and $\text{tr}\{\mathbf{R}_{\mathbf{xx}}^{-1}\}$ can be expressed respectively as

$$\begin{aligned} \text{tr}\{\mathbf{R}_{\mathbf{xx}}\} &= L(\sigma_{x_1}^2 + \sigma_{x_2}^2), \\ \text{tr}\{\mathbf{R}_{\mathbf{xx}}^{-1}\} &= L[1/\sigma_{x_1}^2 + 1/\sigma_{x_2}^2]. \end{aligned}$$

The F-norm condition number of $\mathbf{R}_{\mathbf{xx}}^{1/2}$ is thus given as

$$\begin{aligned} \chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}] &= \|\mathbf{R}_{\mathbf{xx}}^{1/2}\|_F \|\mathbf{R}_{\mathbf{xx}}^{-1/2}\|_F \\ &= \left[\text{tr}\{\mathbf{R}_{\mathbf{xx}}\}\right]^{1/2} \left[\text{tr}\{\mathbf{R}_{\mathbf{xx}}^{-1}\}\right]^{1/2} \\ &= L \frac{(\sigma_{x_1}^2 + \sigma_{x_2}^2)}{\sqrt{(\sigma_{x_1}^2 \sigma_{x_2}^2)}}, \end{aligned} \quad (5.153)$$

while the E-norm condition number of $\mathbf{R}_{\mathbf{xx}}^{1/2}$ can be written explicitly as

$$\begin{aligned}\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}] &= \|\mathbf{R}_{\mathbf{xx}}^{1/2}\|_E \|\mathbf{R}_{\mathbf{xx}}^{-1/2}\|_E \\ &= \left[\frac{1}{2L} \text{tr}\{\mathbf{R}_{\mathbf{xx}}\} \right]^{1/2} \left[\frac{1}{2L} \text{tr}\{\mathbf{R}_{\mathbf{xx}}^{-1}\} \right]^{1/2} \\ &= \frac{1}{2} \frac{(\sigma_{x_1}^2 + \sigma_{x_2}^2)}{\sqrt{(\sigma_{x_1}^2 \sigma_{x_2}^2)}}.\end{aligned}\quad (5.154)$$

Hence the relationship between $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ and $\chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ for uncorrelated $x_1(n)$ and $x_2(n)$ is given by

$$\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}] = \frac{1}{2L} \chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}]. \quad (5.155)$$

Comparing (5.153) and (5.154), it can be seen that $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ is independent of L . For the case of $\sigma_{x_1}^2 = \sigma_{x_2}^2 = 1$, $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}] = 1$ while the measure $\chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}] = 2L$ is dependent on the length of the filter. Hence $\chi_F[\mathbf{R}_{\mathbf{xx}}^{1/2}]$ increases with L , while $\chi_E[\mathbf{R}_{\mathbf{xx}}^{1/2}] = 1$, for WGN with uncorrelated $x_1(n)$ and $x_2(n)$ regardless of L .

5.8.6 Additional results with interchannel coherence controlled by adding WGN to channel 2

The use of the non-linear processing [52] defined by (4.46a) and (4.46b) for varying the interchannel coherence was chosen in Section 5.6.1 since this is a well-established method of reducing interchannel coherence without degrading the stereo image and signal quality significantly as can be found in existing literature. In this section, an additional simulation result is presented to verify that the analysis as presented in Section 5.4 is also valid for various interchannel coherences controlled by adding a zero mean uncorrelated WGN to one of the two channels as shown in Fig. 5.15.

In this simulation, the impulse responses of both the transmission and receiving rooms are recorded at 16 kHz sampling rate. Signals $x_1(n)$ and $x_2(n)$ are generated by convolving a zero mean unit variance WGN source with $g_1(n)$ and $g_2(n)$. A zero mean uncorrelated WGN sequence is added with various SNRs to $x_2(n)$ giving $\tilde{x}_2(n)$ which serves to vary the interchannel coherence. The level of decorrelating noise added to $x_2(n)$ is denoted by SNR_d . The received signals are obtained by convolving $x_1(n)$ and $\tilde{x}_2(n)$ with receiving room impulse responses. With reference to Fig. 4.1, and defining

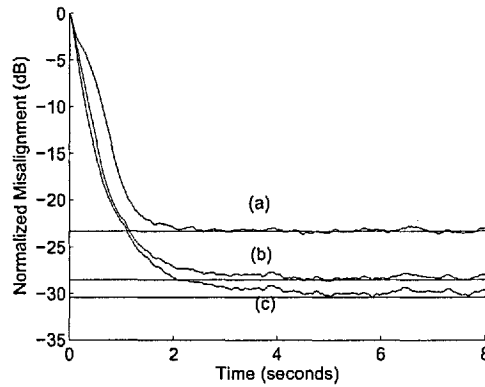


Figure 5.15: Effect of interchannel coherence on misalignment where interchannel coherence is controlled by addition of WGN to $x_2(n)$ giving interchannel coherences of (a) 0.90, (b) 0.67 and (c) 0.35 [$L_T = L_R = 1024$, $L = 1024$, $f_s = 16$ kHz, $\lambda = [1 - 1/(3L)]^L$, SNR = 30 dB].

\otimes as convolution operator, an uncorrelated zero mean WGN sequence $w(n)$ is added to $x_1(n) \otimes h_1(n) + \check{x}_2(n) \otimes h_2(n)$ to obtain an SNR of 30 dB. In this simulation, filter input sequences $x_1(n)$ and $\check{x}_2(n)$ are used to verify the analysis presented in Section 5.4. Figure 5.15 (a), (b) and (c) show normalized misalignments with SNR_d of (a) 10, (b) 0 and (c) -20 dB giving interchannel coherences of (a) 0.90, (b) 0.67 and (c) 0.35 respectively. It can be seen that as the SNR_d reduces, more uncorrelated noise is added to $x_2(n)$ hence reducing the interchannel coherences giving a good misalignment performance. Theoretical normalized misalignments are computed using (5.120) and (5.123b) and are plotted as horizontal lines for various interchannel coherences. Note that the analysis presented in Section 5.4 is also valid using this method of decorrelation. In fact, although the non-linearity control factor β is used to vary the interchannel coherences in Section 5.6.1, the analysis presented does not make any assumptions about the methods of achieving this interchannel coherence variation.

5.8.7 Frequency-domain algorithms

Table 5.1: Data sectioning matrices

If total blocks $\mathcal{K} = 1$, then $N = L$

$$\mathbf{W}_{N \times 2N}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} & \mathbf{0}_{N \times N} \end{bmatrix}_{N \times 2N}$$

$$\mathbf{W}_{2N \times N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} \\ \mathbf{I}_{N \times N} \end{bmatrix}_{2N \times N}$$

$$\mathbf{W}_{2N \times N}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix}_{2N \times N}$$

$$\mathbf{W}_{2N \times 2N}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \end{bmatrix}_{2N \times 2N}$$

$$\mathbf{W}_{2N \times 2N}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \end{bmatrix}_{2N \times 2N}$$

$$\mathbf{G}_{2N \times 2N}^{01} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{01} \mathbf{F}_{2N \times 2N}^{-1}$$

$$\mathbf{G}_{2N \times 2N}^{10} = \mathbf{F}_{2N \times 2N} \mathbf{W}_{2N \times 2N}^{10} \mathbf{F}_{2N \times 2N}^{-1}$$

Table 5.2: The FLMS algorithm [22] [11]

$$\begin{aligned}
0 < \mu &\leq 1 \\
\lambda &= [1 - 1/(3L)]^L \\
\mathbf{X}(m) &= [x(mL - L) \ x(mL - L + 1) \ \dots \ x(mL + L - 1)]^T \\
\mathbf{X}(m) &= \mathbf{F}_{2L \times 2L} \mathbf{X}(m) \\
\mathbf{D}(m) &= \text{diag}\{\mathbf{X}(m)\} \\
\hat{\mathbf{Y}}(m) &= \mathbf{G}_{2L \times 2L}^{01} \mathbf{D}(m) \hat{\mathbf{h}}(m - 1) \\
\mathbf{E}(m) &= \mathbf{Y}(m) - \hat{\mathbf{Y}}(m) \\
\mathcal{P}_{\text{FLMS}}(m) &= \lambda \mathcal{P}_{\text{FLMS}}(m - 1) + (1 - \lambda) \mathbf{D}^*(m) \mathbf{D}(m) \\
\hat{\mathbf{h}}(m) &= \hat{\mathbf{h}}(m - 1) + 2\mu(1 - \lambda) \mathbf{G}_{2L \times 2L}^{10} \mathbf{D}^*(m) \times \\
&\quad [\mathcal{P}_{\text{FLMS}}(m) + \delta_{\text{FLMS}} \mathbf{I}_{2L \times 2L}]^{-1} \mathbf{E}(m)
\end{aligned}$$

Table 5.3: The MDF algorithm [23] [11]

$$\begin{aligned}
0 < \mu &\leq 1 \\
\lambda &= [1 - 1/(3L)]^N \\
k &= 0, 1, \dots, K - 1 \\
\mathbf{D}(m - k) &= \text{diag}\{\text{FFT}\{x(mN - kN - N) \ \dots \ x(mN - kN + N - 1)\}\} \\
\hat{\mathbf{y}}(m) &= \mathbf{G}_{2N \times 2N}^{01} \sum_{k=0}^{K-1} \mathbf{D}(m - k) \hat{\mathbf{h}}_k(m - 1) \\
\mathbf{e}(m) &= \mathbf{y}(m) - \hat{\mathbf{y}}(m) \\
\mathcal{P}_{\text{MDF}}(m) &= \lambda \mathcal{P}_{\text{MDF}}(m - 1) + (1 - \lambda) \mathbf{D}^*(m) \mathbf{D}(m) \\
\hat{\mathbf{h}}_k(m) &= \hat{\mathbf{h}}_k(m - 1) + 2\mu(1 - \lambda) \mathbf{G}_{2N \times 2N}^{10} \mathbf{D}^*(m - k) \times \\
&\quad [\mathcal{P}_{\text{MDF}}(m) + \delta_{\text{MDF}} \mathbf{I}_{2L \times 2L}]^{-1} \mathbf{e}(m)
\end{aligned}$$

Table 5.4: The two-channel frequency-domain adaptive algorithm [11]

for the j^{th} channel	
$0 \ll \lambda < 1$	
$0 < \mu \leq 2$	
μ'	$= \mu(1 - \lambda)$
$\mathbf{X}_j(m)$	$= [x_j(mL - L) \ x_j(mL - L + 1) \ \dots \ x_j(mL + L - 1)]^T$
$\underline{\mathbf{X}}_j(m)$	$= \mathbf{F}_{2L \times 2L} \mathbf{X}_j(m)$
$\underline{\mathbf{D}}_j(m)$	$= \text{diag}\{\underline{\mathbf{X}}_j(m)\}$
$\widehat{\mathbf{S}}_{jr}(m)$	$= \lambda \widehat{\mathbf{S}}_{jr}(m - 1) + (1 - \lambda) \underline{\mathbf{D}}_j^*(m) \underline{\mathbf{D}}_j(m), \ j, r = 1, 2$
$ \widehat{\Gamma}(m) ^2$	$= [\widehat{\mathbf{S}}_{11}(m) \widehat{\mathbf{S}}_{22}(m)]^{-1} \widehat{\mathbf{S}}_{21}(m) \widehat{\mathbf{S}}_{12}(m)$
$\widehat{\mathbf{S}}_j(m)$	$= \widehat{\mathbf{S}}_{jj}(m) [\mathbf{I}_{2L \times 2L} - \widehat{\Gamma}(m) ^2], \ j = 1, 2$
$\underline{\mathbf{K}}_1(m)$	$= \widehat{\mathbf{S}}_1^{-1}(m) [\underline{\mathbf{D}}_1^*(m) - \widehat{\mathbf{S}}_{12}(m) \widehat{\mathbf{S}}_{22}^{-1}(m) \underline{\mathbf{D}}_2^*(m)]$
$\underline{\mathbf{K}}_2(m)$	$= \widehat{\mathbf{S}}_2^{-1}(m) [\underline{\mathbf{D}}_2^*(m) - \widehat{\mathbf{S}}_{21}(m) \widehat{\mathbf{S}}_{11}^{-1}(m) \underline{\mathbf{D}}_1^*(m)]$
$\underline{\mathbf{e}}(m)$	$= \underline{\mathbf{y}}(m) - \mathbf{G}_{2L \times 2L}^{01} [\underline{\mathbf{D}}_1(m) \widehat{\mathbf{h}}_1(m - 1) + \underline{\mathbf{D}}_2(m) \widehat{\mathbf{h}}_2(m - 1)]$
$\widehat{\mathbf{h}}_j(m)$	$= \widehat{\mathbf{h}}_j(m - 1) + \mu' \underline{\mathbf{K}}_j(m) \underline{\mathbf{e}}(m), \ j = 1, 2$

Table 5.5: The XM-FLMS algorithm

for the j^{th} channel,	
$0 < \mu \leq 1$	
$M = 0.5L$	
λ	$= [1 - 1/(3L)]^L$
$\mathbf{X}_j(m)$	$= [x_j(mL - L) \ x_j(mL - L + 1) \ \dots \ x_j(mL + L - 1)]^T$
$\mathbf{X}_{b,j}(m)$	$= [x_j(mL) \ x_j(mL + 1) \ \dots \ x_j(mL + L - 1)]^T$
$\mathbf{p}(m)$	$= \mathbf{X}_{b,1}(m) - \mathbf{X}_{b,2}(m) $
$\mathbf{q}_j(m)$	$= [q_{j,0}(m) \ q_{j,1}(m) \ \dots \ q_{j,L-1}(m)]^T$
$\mathbf{Q}_j(m)$	$= \text{diag}\{\mathbf{q}_j(m)\}$
$\tilde{\mathbf{X}}_{b,j}(m)$	$= \mathbf{Q}_j(m)\mathbf{X}_{b,j}(m)$
$\tilde{\mathbf{X}}_j(m)$	$= [\tilde{\mathbf{X}}_{b,j}(m-1) \ \tilde{\mathbf{X}}_{b,j}(m)]^T$
$\underline{\mathbf{X}}_j(m)$	$= \mathbf{F}_{2L \times 2L} \mathbf{X}_j(m)$
$\tilde{\underline{\mathbf{X}}}_j(m)$	$= \mathbf{F}_{2L \times 2L} \tilde{\mathbf{X}}_j(m)$
$\underline{\mathbb{D}}_j(m)$	$= \text{diag}\{\underline{\mathbf{X}}_j(m)\}$
$\tilde{\underline{\mathbb{D}}}_j(m)$	$= \text{diag}\{\tilde{\underline{\mathbf{X}}}_j(m)\}$
$\hat{\underline{\mathbf{Y}}}(m)$	$= \mathbf{G}_{2L \times 2L}^{01} \sum_{j=1}^2 \underline{\mathbb{D}}_j(m) \hat{\mathbf{h}}_j(m-1)$
$\underline{\mathbf{E}}(m)$	$= \underline{\mathbf{Y}}(m) - \hat{\underline{\mathbf{Y}}}(m)$
$\mathcal{P}_{\text{FLMS}}(m)$	$= \lambda \mathcal{P}_{\text{FLMS}}(m-1) + (1 - \lambda) \sum_{j=1}^2 \underline{\mathbb{D}}_j^*(m) \underline{\mathbb{D}}_j(m)$
$\hat{\mathbf{h}}_j(m)$	$= \hat{\mathbf{h}}_j(m-1) + 2\mu(1 - \lambda) \mathbf{G}_{2L \times 2L}^{10} \tilde{\underline{\mathbb{D}}}_j^*(m) \times$ $[\mathcal{P}_{\text{FLMS}}(m) + \delta_{\text{FLMS}} \mathbf{I}_{2L \times 2L}]^{-1} \underline{\mathbf{E}}(m)$
$q_{1,u}(m)$	$= \begin{cases} 1, & p_u(m) \in \{M \text{ maxima of } \mathbf{p}(m)\} \\ 0, & \text{otherwise} \end{cases}$
$q_{2,v}(m)$	$= \begin{cases} 1, & p_v(m) \in \{M \text{ minima of } \mathbf{p}(m)\} \\ 0, & \text{otherwise} \end{cases}$

Table 5.6: The XM-FLMS α algorithm

for the j^{th} channel,	
$0 < \mu \leq 1$	
$M = 0.5L$	
λ	$= [1 - 1/(3L)]^L$
$\mathbf{X}_{\alpha,j}(m)$	$= [x_j(mL/\alpha - L) \ x_j(mL/\alpha - L + 1) \ \dots \ x_j(mL/\alpha + L - 1)]^T$
$\mathbf{p}(m)$	$= \mathbf{X}_{\alpha,1}(m) - \mathbf{X}_{\alpha,2}(m) $
$\mathbf{q}_j(m)$	$= [q_{j,0}(m) \ q_{j,1}(m) \ \dots \ q_{j,2L-1}(m)]^T$
$\mathbf{Q}_j(m)$	$= \text{diag}\{\mathbf{q}_j(m)\}$
$\tilde{\mathbf{X}}_{\alpha,j}(m)$	$= \mathbf{Q}_j(m)\mathbf{X}_{\alpha,j}(m)$
$\underline{\mathbf{X}}_{\alpha,j}(m)$	$= \mathbf{F}_{2L \times 2L}\mathbf{X}_{\alpha,j}(m)$
$\tilde{\underline{\mathbf{X}}}_{\alpha,j}(m)$	$= \mathbf{F}_{2L \times 2L}\tilde{\mathbf{X}}_{\alpha,j}(m)$
$\underline{\mathbb{D}}_{\alpha,j}(m)$	$= \text{diag}\{\underline{\mathbf{X}}_{\alpha,j}(m)\}$
$\tilde{\underline{\mathbb{D}}}_{\alpha,j}(m)$	$= \text{diag}\{\tilde{\underline{\mathbf{X}}}_{\alpha,j}(m)\}$
$\hat{\mathbf{Y}}_{\alpha}(m)$	$= \mathbf{G}_{2L \times 2L}^{01} \sum_{j=1}^2 \underline{\mathbb{D}}_{\alpha,j}(m)\hat{\mathbf{h}}_j(m-1)$
$\underline{\mathbf{E}}_{\alpha}(m)$	$= \underline{\mathbf{Y}}_{\alpha}(m) - \hat{\mathbf{Y}}_{\alpha}(m)$
$\mathcal{P}_{\text{FLMS}}(m)$	$= \lambda \mathcal{P}_{\text{FLMS}}(m-1) + (1-\lambda) \sum_{j=1}^2 \underline{\mathbb{D}}_{\alpha,j}^*(m)\underline{\mathbb{D}}_{\alpha,j}(m)$
$\hat{\mathbf{h}}_j(m)$	$= \hat{\mathbf{h}}_j(m-1) + 2\mu(1-\lambda)\mathbf{G}_{2L \times 2L}^{10}\tilde{\underline{\mathbb{D}}}_{\alpha,j}^*(m) \times$ $[\mathcal{P}_{\text{FLMS}}(m) + \delta_{\text{FLMS}}\mathbf{I}_{2L \times 2L}]^{-1}\underline{\mathbf{E}}_{\alpha}(m)$
$q_{1,u}(m)$	$= \begin{cases} 1, & p_u(m) \in \{M \text{ maxima of } \mathbf{p}(m)\} \\ 0, & \text{otherwise} \end{cases}$
$q_{2,v}(m)$	$= \begin{cases} 1, & p_v(m) \in \{M \text{ minima of } \mathbf{p}(m)\} \\ 0, & \text{otherwise} \end{cases}$

Chapter 6

Discussion and Conclusions

*Problems cannot be solved at the same
level of awareness that created them.*

Albert Einstein (1879-1955)

6.1 Summary

In this thesis, a class of time- and frequency-domain selective-tap algorithms were developed and analyzed for single channel and stereophonic AEC applications. In Chapter 2, having reviewed partial update algorithms, the MMax tap selection [21] was extended to the affine projection algorithm (MMax-AP). Using normal equations, the MMax recursive least squares algorithm (MMax-RLS) for single channel AEC was developed. It was noted that MMax-NLMS, being a data-dependent partial update algorithm, outperforms the SPU-NLMS [20], Periodic-NLMS and Sequential-NLMS algorithms [19]. It was also noted that the rate of convergence of adaptive algorithms employing MMax tap selection degrades gracefully with reducing size of tap selection M . Convergence for both WGN and speech input sequences were shown for MMax-AP and MMax-RLS through simulations. Although all filter coefficients were selected for adaptation, for an example case of $L = 1024$ and $M = 512$, it has been shown that the number of operations required by MMax-AP and MMax-RLS employing the SORTLINE algorithm is approximately 75.7% and 62.5% that of the number for AP and RLS respectively.

The steady-state misalignment analysis for a class of fully updated and their cor-

responding MMax selective-tap algorithms were presented, in Chapter 3, under both time-varying and time-invariant unknown system conditions modelled using the modified Markov model [77]. Under time-invariant system conditions, the steady-state normalized misalignment for MMax-NLMS and MMax-AP were found to be independent of M while the same is not true for MMax-RLS. For a time-varying system, the performance of MMax-based algorithms in terms of steady-state misalignment degrades with increasing time-variation. This degradation is proportional to ψ for both MMax-NLMS and MMax-RLS algorithms. For $0.5L \leq M < L$, the increase in ψ is insignificant and as a consequence, the degradation in steady-state misalignment performance is negligible with reducing M . This property was exploited for the XM tap selection, which was then deployed in SAEC algorithms such as presented in [30] [26]. It was additionally shown for NLMS and MMax-NLMS that, under time-varying unknown system conditions, there exist an optimal step-size given by (3.33) and (3.65) respectively. This optimal step-size jointly maximizes the performances of the algorithms in terms of low misalignment and high convergence rate. Simulation results were presented and have shown to verify that the analysis accurately describes the performances of the algorithms. This analysis enables a judicious trade-off between the computational savings of partial update schemes and their tracking performance.

In Chapter 4, a novel tap selection approach to reduce the interchannel coherence for stereophonic acoustic echo cancellation (SAEC) was proposed. It was shown that to reduce the degradation in convergence rate due to tap selection, the proposed M-ratio \mathcal{M} has to be maximized. It was noted that the MMax tap selection imposed on SAEC adaptive algorithms will not achieve sufficient convergence performance since, due to the high interchannel coherence, same tap-indices will be selected for both channels. As a proof of concept, a joint optimization problem was formulated by maximizing the MMax-criterion and minimizing the interchannel coherence under the control of tap selection. The resultant exhaustive search technique has shown to achieve good convergence performance over the fully-update NLMS algorithm when taps corresponding to tap-input combinations having the highest \mathcal{M} subjected to exclusivity are selected for adaptation. The exclusive-maximum (XM) tap selection algorithm was proposed which efficiently selects taps corresponding to tap-inputs maximizing \mathcal{M} subjected to exclusivity. This XM tap selection was extended to the NLMS, AP and RLS algorithms and when employed

with the non-linear (NL) preprocessor, higher rates of convergence in the range of approximately 3 to 7 dB are exhibited compared to their respective algorithms employing NL preprocessor alone. Alternatively, the distortion factor β for the XMNL-based algorithms can be reduced to achieve the same convergence rates as algorithms that employ the NL-preprocessor alone.

In Chapter 5, frequency-domain analysis was presented and frequency-domain algorithms employing tap selection were developed. Utilizing the E-norm condition number [111], the link between interchannel coherence and the conditioning of the two-channel correlation matrix \mathbf{R}_{xx} was established. Employing this relationship, an insight of how the XM tap selection reduces the interchannel coherence was presented. It was further shown how the misalignment performances of SAEC algorithms were improved through the better conditioning of \mathbf{R}_{xx} . The measure \mathcal{M} has been shown to reduce significantly if subselection is employed on the frequency-domain tap-input vector hence the rate of convergence of frequency-domain SAEC algorithms is reduced significantly. Consequently, for the proposed frequency-domain algorithms, decisions on tap selection were based on time-domain tap-input vectors. Two frequency-domain selective-tap algorithms were proposed; one employing the 50% overlapping factor (XMNL-FLMS) and the other employing an arbitrary overlapping factor (XMNL-FLMS α) controlled by $\alpha > 1$. Simulation results were presented and the proposed XMNL-based algorithms were shown to achieve improved convergence over algorithms that employ the NL preprocessor alone.

6.2 Conclusions

In this work selective-tap algorithms employing the MMax tap selection for acoustic echo cancellation (AEC) were developed and analyzed. It has been shown, for single channel AEC, that although all taps are updated, reduced computational complexity can be achieved in selective-tap algorithms compared to their corresponding fully updated algorithms. Analysis and simulations presented have shown that selective-tap algorithms suffer from degradation in terms of both convergence rate and steady-state misalignment under time-varying unknown system conditions. Analytical results have shown to be accurate to within an estimation error of approximately 0.1 to 0.2 dB compared with simulated results hence validating the analysis presented. More importantly, it has been found that

for the case of $M = 0.5L$, the degradation in performance of selective-tap algorithms is insignificant. Specifically, this degradation is in the range of approximately 0.5 to 1 dB in terms of steady-state misalignment for the single channel MMax-based algorithms under time-varying unknown system conditions. This motivates the novel application of such selective-tap algorithms using the exclusive-maximum (XM) tap selection for reducing interchannel coherence so as to achieve good convergence performance in stereophonic acoustic echo cancellation (SAEC). Although the main aim of such a deployment for selective-tap algorithms was not the reduction in computational complexity as in the single channel AEC case, reduced complexity compared to the fully updated algorithms can nevertheless be seen. It has been shown that the overall performance increase for SAEC in the region of 3 to 7 dB can be obtained with a computational load of 76% that of NL-NLMS using the selective-tap XMNL-NLMS algorithm. This is due to the additional decorrelating effect brought about by the exclusive tap selection. For the XMNL-AP and XMNL-RLS algorithms, with approximately the same improvement in convergence rate compared to NL-AP and NL-RLS, the reduction in computational load is approximately 75.7% and 62.5% that of the fully updated NL-AP and NL-RLS algorithms respectively. As a consequence, such selective-tap algorithms can be applied to both single and two-channel system identification applications such as AEC considered in this thesis.

6.3 Future work

The convergence properties of tap selection adaptive algorithms were shown in this thesis. These algorithms can be applied to reduce interchannel coherence so as to achieve good misalignment performance for adaptive algorithms. The proposed XM selective-tap algorithms can be employed as a platform for other SAEC algorithms (time- or frequency-domain) to enhance their convergence performances. Subband approaches employing XM tap selection for SAEC can also be explored.

Typically, adaptive algorithms for AEC perform the role of system identification which aim to model the receiving room's impulse response. The use of XM tap selection algorithms can also be considered for use in acoustic feedback cancellation (AFC) applications in multi-channel closed-loop systems. Such systems have been deployed in, for example, commercial and military applications such as medical hearing-aids and automotive cabin

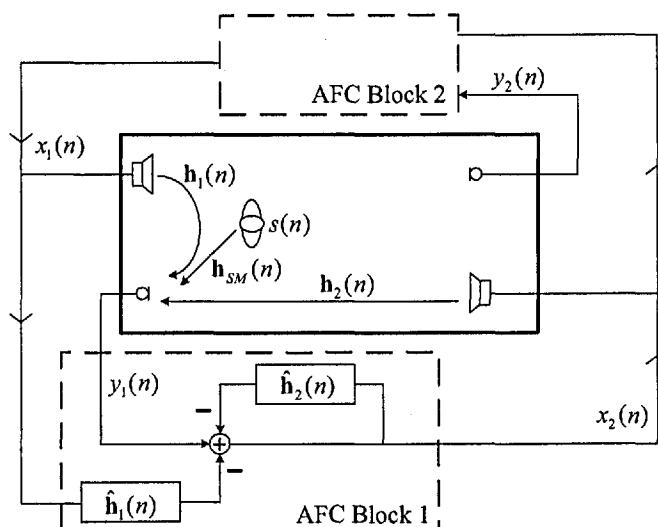


Figure 6.1: Schematic of an in-car communication system.

communications systems such as depicted in Fig. 6.1. It has been shown [16] that one of the main problems in such closed-looped system, similar to the SAEC case, is the high interchannel coherence between $x_1(n)$ and $x_2(n)$. It has been proposed in [16] that the non-linear preprocessor can be employed in such applications, similar to the SAEC case. A possible extension of this work is thus to employ the XM tap selection to further reduce the interchannel coherence in applications where highly correlated interferers exist such as described above.

6.4 List of publications arising directly from this thesis

- Book Chapter

1. P. A. Naylor and A. W. H. Khong, "Selective-tap adaptive algorithms for echo cancellation," in *Selected Methods for Acoustic Echo and Noise Control*, E. Hänsler and G. Schmidt, ed.. Springer, to appear 2006.

- Journals

1. A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive filtering with performance analysis for non-stationary system identification," *IEEE Trans. Speech Audio Processing*, submitted Nov. 2005.

2. A. W. H. Khong, J. Benesty and P. A. Naylor, "Stereophonic acoustic echo cancellation: Analysis of the misalignment in the frequency domain," *IEEE Signal Processing Lett.*, to appear Jan. 2006.
 3. A. W. H. Khong and P. A. Naylor, "Stereophonic acoustic echo cancellation employing selective-tap adaptive algorithms," *IEEE Trans. Speech Audio Processing*, to appear Jul. 2006.
 4. A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive algorithms in the solution of the nonuniqueness problem for stereophonic acoustic echo cancellation," *IEEE Signal Processing Lett.*, vol. 12, no. 4, pp. 269-272, Apr. 2005.
- Conference proceedings
 1. A. W. H. Khong, J. Benesty and P. A. Naylor, "Effect of interchannel coherence on conditioning and misalignment performance for stereo acoustic echo cancellation," *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, submitted Oct. 2005.
 2. A. W. H. Khong and P. A. Naylor, "Frequency domain adaptive algorithms for stereophonic acoustic echo cancellation employing tap selection," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Sep. 2005, pp. 141-144.
 3. A. W. H. Khong and P. A. Naylor, "A family of selective-tap algorithms for stereo acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, vol. 3, Mar. 2005, pp. 133-136.
 4. P. A. Naylor and A. W. H. Khong, "Affine projection and recursive least squares adaptive filters employing partial updates," in *Proc. Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 950-954.
 5. A. W. H. Khong and P. A. Naylor, "Reducing inter-channel coherence in stereophonic acoustic echo cancellation using partial update adaptive filters," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2004, pp. 405-408.
 6. A. W. Khong and P. A. Naylor, "The use of partial update scheme to reduce inter-channel coherence in adaptive stereophonic acoustic echo cancellation," in

Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC), Sep. 2003, pp. 59-62.

7. A. W. Khong and P. A. Naylor, "A partial-update adaptive algorithm for stereophonic acoustic echo cancellation," *presented in young speech researchers meeting, University College London*, Apr. 2003.

6.5 Other publications

1. R. Ahmad, A. W. H. Khong and P. A. Naylor, "Proportionate frequency domain adaptive algorithms for blind channel identification," *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (ICASSP)*, accepted for publication, 2006.
2. A. W. Khong, J. Benesty and P. A. Naylor, "An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2005.

Bibliography

- [1] S. Weinstein, "Echo cancellation in the telephone network," *IEEE Commun. Mag.*, vol. 15, no. 1, pp. 8–15, Jan. 1977.
- [2] M. M. Sondhi and D. A. Berkley, "Silencing echoes on the telephone network," in *Proc. IEEE*, vol. 68, Aug. 1980, pp. 948–963.
- [3] P. T. Brady and G. K. Helder, "Echo suppressor design in telephone communications," *Bell Syst. Tech. J.*, vol. 42, no. 6, pp. 2893–2917, Nov. 1963.
- [4] M. M. Sondhi and A. J. Presti, "A self-adaptive echo canceler," *Bell Sys. Tech. J.*, vol. 45, pp. 1851–1854, Dec. 1966.
- [5] M. M. Sondhi, "An adaptive echo canceller," *Bell Sys. Tech. J.*, vol. 46, no. 3, pp. 497–511, Mar. 1967.
- [6] A. Miura, S. Kobayashi, R. Sato, and K. Nagata, "A blockless echo suppressor," *IEEE Trans. Commun.*, vol. 17, no. 4, pp. 489–495, Aug. 1969.
- [7] D. L. Duttweiler, "A twelve-channel digital echo canceler," *IEEE Trans. Commun.*, vol. 26, no. 5, pp. 647–653, May. 1978.
- [8] D. L. Duttweiler and Y. S. Chen, "A single-chip VLSI echo canceler," *Bell Sys. Tech. J.*, vol. 59, no. 2, Feb. 1980.
- [9] D. A. Berkley and O. M. M. Mitchell, "Seeking the ideal in 'hands-free' telephony," *Bell Laboratories Record*, vol. 52, pp. 318–325, 1974.
- [10] E. Hänsler, "The hands-free telephone problem- an annotated bibliography," *Signal Processing*, vol. 27, no. 3, pp. 259–271, Jun. 1992.

- [11] J. Benesty, T. Gänslér, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Springer, 2001.
- [12] E. Hänsler, "The hands-free telephone problem," in *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. 4, 1992, pp. 1914–1917.
- [13] R. Botros, O. Abdel-Alim, and P. Damaske, "Stereophonic speech teleconferencing," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 11, Apr. 1986, pp. 1321–1324.
- [14] M. M. Sondhi, D. R. Morgan, and J. L. Hall, "Stereophonic acoustic echo cancellation- An overview of the fundamental problem," *IEEE Signal Processing Lett.*, vol. 2, no. 8, pp. 148–151, Aug. 1995.
- [15] C. Breining, "Control of a hands-free telephone set," *Signal Processing*, vol. 61, pp. 131–143, 1997.
- [16] G. Schmidt, "Applications of acoustic echo control- an overview," in *Proc. Eur. Signal Process. Conf.*, 2004, pp. 9–16.
- [17] E. Hänsler, "Hands-free telephones- joint control of echo cancellation and postfiltering," *Signal Processing*, vol. 80, no. 11, pp. 2295–2305, Nov. 2000.
- [18] S. J. Park, C. G. Cho, C. Lee, and D. H. Youn, "Integrated echo and noise canceler for hands-free applications," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 3, pp. 188–195, Mar. 2002.
- [19] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 3, pp. 209–216, Mar. 1997.
- [20] K. Dogancay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 8, pp. 762–769, Aug. 2001.
- [21] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1421–1424, 1999.
- [22] E. R. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 474–475, 1980.

- [23] J. S. Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 373–376, Feb. 1990.
- [24] P. A. Naylor and A. W. H. Khong, "Affine projection and recursive least squares adaptive filters employing partial updates," in *Proc. Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 950–954.
- [25] —, "Selective-tap adaptive algorithms for echo cancellation," in *Selected methods for acoustic echo and noise control*, E. Hänsler and G. Schmidt, Eds. Springer, 2006, to appear.
- [26] A. W. H. Khong and P. A. Naylor, "Stereophonic acoustic echo cancellation employing selective-tap adaptive algorithms," *IEEE Trans. Speech Audio Processing*, July 2006, to appear.
- [27] —, "Selective-tap adaptive algorithms in the solution of the non-uniqueness problem for stereophonic acoustic echo cancellation," *IEEE Signal Processing Lett.*, vol. 12, no. 4, pp. 269–272, Apr. 2005.
- [28] —, "The use of partial update scheme to reduce inter-channel coherence in adaptive stereophonic acoustic echo cancellation," in *Proc. International Workshop on Acoustic Echo and Noise Control*, Sep. 2003, pp. 59–62.
- [29] —, "Reducing inter-channel coherence in stereophonic acoustic echo cancellation using partial update adaptive filters," in *Proc. Eur. Signal Process. Conf.*, Sep. 2004, pp. 405–408.
- [30] —, "A family of selective-tap algorithms for stereo acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, Mar. 2005, pp. 133–136.
- [31] A. W. H. Khong, J. Benesty, and P. A. Naylor, "Stereophonic acoustic echo cancellation: Analysis of the misalignment in the frequency domain," *IEEE Signal Processing Lett.*, vol. 13, no. 1, pp. 33–36, Jan. 2006.
- [32] A. W. H. Khong and P. A. Naylor, "Frequency domain adaptive algorithms for stereophonic acoustic echo cancellation employing tap selection," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, Sep. 2005, pp. 141–144.

- [33] A. W. H. Khong, J. Benesty, and P. A. Naylor, "An improved proportionate multi-delay block adaptive filter for packet-switched network echo cancellation," in *Proc. European Signal Processing Conference*, Sep. 2005.
- [34] B. B. Farhang, *Adaptive Filters: Theory and Applications*. Chichester: Wiley, 1998.
- [35] S. Haykin, *Adaptive Filter Theory*, 4th ed., ser. Information and System Science. Prentice Hall, 2002.
- [36] R. Martin and J. Alenhöner, "Coupled adaptive filters for acoustic echo control and noise reduction," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 5, May 1995, pp. 3043–3046.
- [37] V. Turbin, A. Gilloire, and P. Scalart, "Comparison of three post-filtering algorithms for residual acoustic echo reduction," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 1, Apr. 1997, pp. 307–310.
- [38] S. Gustafsson, R. Martin, P. Jax, and P. Vary, "A psychoacoustic approach to combined acoustic echo cancellation and noise reduction," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 5, pp. 245–256, Jul. 2002.
- [39] C. Breining, P. Dreiseitel, E. Hänslér, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control: An application of very-high-order adaptive filter," *IEEE Signal Processing Mag.*, vol. 16, no. 4, pp. 42–69, Jul. 1999.
- [40] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: Analysis, experiments, and application to acoustic echo cancellation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1862–1875, Aug. 1992.
- [41] K. A. Lee and S. Gan, "Improving convergence of the NLMS algorithm using constrained subbands updates," *IEEE Signal Processing Lett.*, vol. 11, no. 9, pp. 736–739, Sept. 2004.
- [42] D. R. Morgan and J. C. Thi, "A delayless subband adaptive filter architecture," *IEEE Trans. Signal Processing*, vol. 43, no. 8, pp. 1819–1830, Aug. 1995.
- [43] Y. Bendel, D. Burshtein, O. Shalvi, and E. Weinstein, "Delayless frequency domain acoustic echo cancellation," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 5, pp. 589–597, Jul. 2001.

- [44] A. P. Liavas and P. A. Regalia, "Acoustic echo cancellation: Do IIR models offer better modelling capabilities than their FIR counterparts?" *IEEE Trans. Signal Processing*, vol. 46, no. 9, pp. 2499–2504, Sept. 1998.
- [45] S. L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*. Kluwer Academic Publishers, 2001.
- [46] J. H. Cho, D. R. Morgan, and J. Benesty, "An objective technique for evaluating doubletalk detectors in acoustic echo cancelers," *IEEE Trans. Speech Audio Processing*, vol. 7, no. 6, pp. 718–724, Nov. 1999.
- [47] H. Ye and B. Wu, "A new double-talk detection algorithm based on the orthogonality theorem," *IEEE Trans. Commun.*, vol. 39, no. 11, pp. 1542–1545, Nov. 1991.
- [48] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 2, pp. 168–172, Mar. 2000.
- [49] *ITU-T P.340 Transmission characteristics and speech quality parameters of hands-free terminals*, International Telecommunication Union, 2000.
- [50] *ITU-T G.168 Digital network echo cancellers*, International Telecommunication Union, 2002.
- [51] *ITU-T G.167 Acoustic echo controllers*, International Telecommunication Union, 1993.
- [52] J. Benesty, D. R. Morgan, and M. M. Sondhi, "A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation," *IEEE Trans. Speech Audio Processing*, vol. 6, no. 2, pp. 156–165, Mar. 1998.
- [53] M. G. Bellanger, *Adaptive Digital Filters and Signal Analysis*. Marcel Dekker, 1987.
- [54] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1989.
- [55] E. Parzen, *Stochastic Processes*. Holden-Day, Inc, 1962.
- [56] B. Widrow, "Adaptive filters," in *Aspects of network and system theory*, R. Kalman and N. DeClaris, Eds. Holt, Rinehart and Winston, 1971, pp. 563–587.

- [57] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," in *Proc. IEEE*, vol. 63, no. 12, 1975, pp. 1692–1716.
- [58] B. Widrow, "Thinking about thinking: the discovery of the LMS algorithm," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 100–106, Jan. 2005.
- [59] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *Proc. Institute of Radio Enginneers Western Electronic Show and Convention (IRE WESCON Conv. Rec.)*, no. 4, 1960, pp. 96–104.
- [60] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*. Prentice Hall, 1984.
- [61] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial update NLMS algorithms with data-selective updating," *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 938–949, 2004.
- [62] 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 Lett.*, vol. 5, no. 5, pp. 111–114, May 1998.
- [63] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, no. 3, pp. 282–287, Jun. 1967.
- [64] S. C. Douglas, "A family of normalized LMS algorithms," *IEEE Signal Processing Lett.*, vol. 1, no. 3, pp. 49–51, Mar. 1994.
- [65] ———, "Analysis and implementation of the MAX-NLMS adaptive filter," in *Proc. Twenty-Ninth Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1995, pp. 659–663.
- [66] T. Aboulnasr and K. Mayyas, "Selective coefficient update of gradient-based adaptive algorithms," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, 1997, pp. 1929–1932.
- [67] K. Ozeki and T. Umeda, "An adaptive filtering using an orthogonal projection to an affine subspace and properties," *Electronics and Communications in Japan*, vol. 67-A, no. 5, 1984.

- [68] S. L. Gay, "Affine projection algorithms," in *Least-mean-square adaptive filters*, S. Haykin and B. Widrow, Eds. John Wiley & Sons Inc., 2003, ch. 7, pp. 241–291.
- [69] A. S. Householder, *The Theory of Matrices in Numerical Analysis*. Waltham, MA: Blaisdell, 1964.
- [70] I. Pitas, "Fast algorithms for running ordering and max/min calculation," *IEEE Trans. Circuits Syst.*, vol. 36, no. 6, pp. 795–804, Jun. 1989.
- [71] P. A. Naylor and W. Sherliker, "A short-sort M-max NLMS partial update adaptive filter with applications to echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 5, 2003, pp. 373–376.
- [72] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, Apr. 1979.
- [73] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and non-stationary learning characteristics of the LMS adaptive filter," in *Proc. IEEE*, vol. 64, no. 8, 1976, pp. 1151–1162.
- [74] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Trans. Inform. Theory*, vol. 30, no. 2, pp. 211–221, 1984.
- [75] E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 5, pp. 1087–1110, 1986.
- [76] O. Macchi, "Optimization of adaptive identification for time-varying filters," *IEEE Trans. Automat. Contr.*, vol. 31, no. 3, pp. 283–287, 1986.
- [77] N. J. Bershad, S. McLaughlin, and C. F. N. Cowan, "Performance comparison of RLS and LMS algorithms for tracking a first order Markov communications channel," in *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. 1, 1990, pp. 266–270.
- [78] W. Sherliker, "Acoustic echo cancellation algorithms with tap selection for non-stationary environments," Ph.D. dissertation, Imperial College London, 2000.

- [79] V. Myllyla and G. Schmidt, "Pseudo-optimal regularization for affine projection," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, 2002, pp. 1917–1920.
- [80] S. G. Sankaran and A. A. (Louis) Beex, "Convergence behaviour of Affine Projection algorithms," *IEEE Trans. Signal Processing*, vol. 48, no. 4, pp. 1086–1096, Apr. 2000.
- [81] A. H. Sayed and V. H. Nascimento, "Energy conservation and the learning ability of lms adaptive filters," in *Least-mean-square adaptive filters*, S. Haykin and B. Widrow, Eds. John Wiley & Sons Inc., 2003, ch. 3, pp. 79–104.
- [82] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*. North-Holland, 1981.
- [83] T. Aboulnasr and K. Mayyas, "MSE analysis of the M-Max NLMS adaptive algorithm," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, 1998, pp. 1669–1672.
- [84] M. Godavarti and A. O. Hero, "Stability bounds on step-size for the partial update LMS algorithm," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, 1999, pp. 1677–1680.
- [85] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. McGraw-Hill Inc., 1991.
- [86] H. Stark and J. W. Woods, *Probability and random processes with applications to signal processing*, 3rd ed. Prentice Hall, 2002.
- [87] A. Barucha-Reid, "Fixed point theorems in probabilistic analysis," *Bull. Amer. Math. Soc.*, vol. 26, pp. 641–657, 1976.
- [88] M. Rupp, "Contraction mapping: an important property in adaptive filters," in *Proc. Sixth IEEE Digital Signal Processing Workshop*, 1994, pp. 273–276.
- [89] J. F. Doherty and R. Porayath, "A robust echo canceler for acoustic environments," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 5, pp. 389–396, May 1997.

- [90] P. Eneroth, S. L. Gay, T. Gänsler, and J. Benesty, "A real-time implementation of a stereophonic acoustic echo canceller," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 5, pp. 513–523, Jul. 2001.
- [91] S. Emura, Y. Haneda, and S. Makino, "Enhanced frequency-domain adaptive algorithm for stereo echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, 2002, pp. 1901–1904.
- [92] K. Mayyas, "Stereophonic acoustic echo cancellation using lattice orthogonalization," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 7, pp. 517–525, Oct. 2002.
- [93] S. Shimauchi and S. Makino, "Stereo projection echo canceller with true echo path estimation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 5, 1995, pp. 3059–3062.
- [94] A. Gilloire and V. Turbin, "Using auditory properties to improve the behaviour of stereophonic acoustic echo cancellers," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 6, 1998, pp. 3681–3684.
- [95] T. Tangsangiumvisai, J. A. Chambers, and A. G. Constantinides, "Time-varying allpass filters using spectral-shaped noise for signal decorrelation in stereophonic acoustic echo cancellation," in *Proc. Int. Conf. Digital Signal Processing*, 2002, pp. 1273–1276.
- [96] J. Benesty, D. R. Morgan, J. L. Hall, and M. M. Sondhi, "Stereophonic acoustic echo cancellation using nonlinear transformations and comb filtering," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 6, 1998, pp. 3673–3676.
- [97] T. Hoya, Y. Loke, J. Chambers, and P. A. Naylor, "Application of the leaky extended LMS algorithm in stereophonic acoustic echo cancellation," *Signal Processing*, vol. 64, pp. 87–91, 1998.
- [98] N. T. Forsyth, J. A. Chambers, and P. A. Naylor, "An alternating fixed-point algorithm for stereophonic acoustic echo cancellation," *IEE Proc. Vision, Image and Signal Processing*, vol. 149, no. 1, pp. 1–9, 2002.

- [99] J. Benesty, P. Duhamèl, and Y. Grenier, "A multichannel affine projection algorithm with applications to multichannel acoustic echo cancellation," *IEEE Signal Processing Lett.*, vol. 3, no. 2, pp. 35–37, Feb. 1996.
- [100] F. Amand, J. Benesty, A. Gilloire, and Y. Grenier, "A fast two-channel projection algorithm for stereophonic acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, May 1996, pp. 949–952.
- [101] A. Sugiyama, A. Hirano, and K. Nakayama, "Acoustic echo cancellation for conference systems," in *Proc. Eur. Signal Process. Conf.*, 2004, pp. 17–20.
- [102] R. F. Baum, "The correlation function of Gaussian noise passed through nonlinear devices," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 448–456, Jul. 1969.
- [103] D. R. Morgan, J. L. Hall, and J. Benesty, "Investigation of several types of nonlinearities for use in stereo acoustic echo cancellation," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 6, pp. 686–696, Sept. 2001.
- [104] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.
- [105] M. Dentino, J. M. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," in *Proc. IEEE*, vol. 66, no. 12, Dec. 1978, pp. 1658–1659.
- [106] S. K. Mitra, *Digital Signal Processing- A Computer-Based Approach*, 2nd ed. McGraw-Hill, 2001.
- [107] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing- Principles, Algorithms and Applications*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 1996.
- [108] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 3, pp. 744–752, 1981.
- [109] D. Mansour and J. A. Gray, "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, no. 5, pp. 726–734, 1982.

- [110] E. Moulines, O. A. Amrane, and Y. Grenier, "The generalized multidelay adaptive filter: Structure and convergence analysis," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 14–28, Jan. 1995.
- [111] J. Benesty and T. Gänslér, "A recursive estimation of the condition number in the RLS algorithm," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 4, 2005, pp. 25–28.
- [112] —, "New insights into the rls algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 3, pp. 331–339, Mar. 2004.
- [113] H. Buchner, J. Benesty, and W. Kellermann, "Generalized multichannel frequency-domain adaptive filtering: efficient realization and application to hands-free speech communication," *Signal Processing*, vol. 85, pp. 549–570, Mar. 2005.
- [114] T. Gänslér and J. Benesty, "New insights into the stereophonic acoustic echo cancellation problem and an adaptive nonlinearity solution," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 5, pp. 257–267, 2002.
- [115] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Mag.*, vol. 9, no. 1, pp. 14–37, Jan. 1992.
- [116] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of complex Fourier series," *Math. Comp.*, vol. 19, pp. 297–301, Apr. 1965.
- [117] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "Historical notes on the fast Fourier transform," *Proc. IEEE*, vol. 55, no. 10, pp. 1675–1677, Oct. 1967.
- [118] R. Gray, "On the asymptotic eigenvalue distribution of toeplitz matrices," *IEEE Trans. Inform. Theory*, vol. 18, no. 6, pp. 725–730, Nov. 1972.
- [119] M. J. Shensa, "The spectral dynamics of evolving LMS adaptive filters," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 4, Apr. 1979, pp. 950–953.
- [120] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. 15, no. 2, pp. 70–73, Jun. 1967.

- [121] J. E. R. Ferrara, "Frequency-domain adaptive filtering," in *Adaptive filters*, C. F. N. Cowen and P. M. Grant, Eds. Prentice-Hall, 1985, ch. 6, pp. 145–179.
- [122] M. Xu and Y. Grenier, "Time-frequency domain adaptive filters," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, May 1989, pp. 1154–1157.
- [123] J. Benesty and D. R. Morgan, "Frequency-domain adaptive filtering revisited, generalization to the multi-channel case, and application to acoustic echo cancellation," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 2, 2000, pp. 789–792.
- [124] P. C. W. Sommen, P. J. van Gerwen, H. J. Kotmans, and A. J. E. M. Janssen, "Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function," *IEEE Trans. Circuits Syst.*, vol. 34, no. 7, pp. 788–798, Jul. 1987.
- [125] J. C. Lee and C. W. Un, "Performance analysis of frequency-domain block LMS adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, no. 2, pp. 173–189, Feb. 1989.
- [126] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 1996.
- [127] G. Strang, *Linear algebra and its applications*, 3rd ed. Thomas Learning, Inc, 1988.