

Joint Time Delay Estimation and Adaptive Filtering Techniques

by

Daniel Boudreau

M. Eng.

Department of Electrical Engineering
McGill University
Montreal, Canada

November 1990

A thesis submitted to the Faculty of Graduate
Studies and Research in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

©Daniel Boudreau, 1990

Abstract

This thesis studies adaptive filters for the case in which the main input signal is not synchronized with the reference signal. The asynchrony is modelled by a time-varying delay. This delay has to be estimated and compensated. This is accomplished by designing and investigating joint delay estimation and adaptive filtering algorithms. First, a joint maximum likelihood estimator is derived for input Gaussian signals. It is used to define a readily implementable joint estimator, composed of an adaptive delay element and an adaptive filter. Next, two estimation criteria are investigated with that structure. The minimum mean squared error criterion is used with a joint steepest-descent adaptive algorithm and with a joint least-mean-square adaptive algorithm. The general convergence conditions of the joint steepest-descent algorithm are derived. The joint LMS algorithm is analysed in terms of joint convergence in the mean and in the mean square. Finally, a joint recursive least squares adaptive algorithm is investigated in conjunction with the exponentially weighted least squares criterion. Experimental results are obtained for these different adaptive algorithms, in order to verify the analyses. The results show that the joint algorithms improve the performance of the conventional adaptive filtering techniques.

Sommaire

Cette thèse examine d'une façon détaillée le problème de synchronisation entre le signal principal et le signal de référence utilisés par un filtre numérique adaptatif. Le manque de synchronisme est représenté par le modèle mathématique d'un délai temporel variable dans le temps. Ce délai doit être estimé et corrigé. Cette tâche est accomplie en concevant et en étudiant différents algorithmes effectuant conjointement une estimation de délai et le filtrage adaptatif. Un estimateur conjoint, basé sur le critère de maximum de vraisemblance, est dérivé en premier lieu en utilisant un signal d'entrée Gaussien. Cet estimateur est utilisé comme base pour définir une forme d'estimateur conjoint facilement applicable, composée d'un délai adaptatif et d'un filtre adaptatif. En second lieu, cette structure est alors étudiée en utilisant deux critères d'estimation. Le critère d'erreur quadratique moyenne est utilisé avec un algorithme adaptatif conjoint à descente maximale et avec un algorithme adaptatif conjoint LMS. Les conditions générales de convergence sont dérivées pour l'algorithme conjoint à descente maximale. L'algorithme conjoint LMS est analysé en termes de convergence des moments du premier et second ordres. Finalement, un algorithme conjoint de moindres carrés récursifs (RLS) à pondération exponentielle est utilisé avec le critère des moindres carrés. Des résultats expérimentaux sont obtenus pour vérifier les dérivations analytiques. Les résultats montrent que les algorithmes conjoints améliorent les performances des techniques conventionnelles de filtrage adaptatif.

Acknowledgments

I would like to sincerely thank my supervisor, Professor Peter Kabal, for his guidance during this research and for his financial assistance provided during conference attendance. My acknowledgments also go to Professor Maier Blostein and to Professor Pierre Bélanger for their advice during my thesis proposal. The review of the thesis manuscript by Dr. David Falconer is a great source of gratification.

The financial assistance provided by the Canadian Department of Communications and the moral support of my colleagues at the Communications Research Centre are greatly appreciated. The computer facilities provided by INRS-Télécommunications helped much the completion of this work.

The companionship provided by the students at McGill and INRS is very much appreciated. The software provided by fellow student Duncan Bees is also gratefully acknowledged.

Finally, special thanks go to my greatest supporters, my wife Josée and my parents. Josée faced the difficult task of supporting long hours of solitude and the changing moods that research work brings. Her love and patience made the completion of this thesis possible. My parents provided all the support and encouragement that is possible to bring. *A vous trois, je dédie cette thèse et tous les efforts que sa rédaction a nécessités. Une partie de votre esprit s'y trouve uni au mien.*

Table of Contents

<i>Abstract</i>	<i>ii</i>
<i>Sommaire</i>	<i>iii</i>
<i>Acknowledgments</i>	<i>iv</i>
<i>Table of Contents</i>	<i>v</i>
<i>List of Figures</i>	<i>ix</i>
<i>List of Symbols and Abbreviations</i>	<i>xiv</i>
Chapter 1 Introduction	1
1.1 Conventional Adaptive Filtering versus Delay Estimation	1
1.2 Conventional Delay Estimation	3
1.3 Conventional Adaptive Transversal Filtering	5
1.4 Thesis Objectives	6
1.5 A Survey of Joint Algorithms Involving Adaptive Filters	7
1.6 Thesis Organization	8
Chapter 2 Joint Time Delay Estimation and Channel Identification	10
2.1 Introduction	10
2.2 The Mathematical Model	10
2.3 The Joint Maximum Likelihood Estimator For a Type I System	12
2.3.1 The Joint ML Estimator for Finite Observation Interval	12
2.3.2 The Joint ML Estimator for Long Observation Interval	15
2.3.3 Discussion	19
2.4 The Joint MMSE and LS Estimators	19
2.4.1 The Sampling Rate Difference Problem in Adaptive Filtering	21
2.4.2 Discussion	24
2.5 Summary	25
Chapter 3 Joint Time Delay Estimation and Adaptive Minimum Mean Squared Error Filtering: The Joint Steepest-Descent Algorithm	26
3.1 Introduction	26
3.2 General MMSE Theory	27
3.2.1 The Mean Squared Error Function	27
3.2.2 Derivative-Based Delay Estimation	32
3.2.3 Discussion	35
3.3 The Joint Steepest-Descent Algorithm	35

3.3.1	Convergence of the Joint SD Algorithm	36
3.3.2	The Delay Tracking Properties of the Algorithm	43
3.4	Application of the Joint SD Algorithm	54
3.4.1	The Function $\ddot{\xi}_{\min}$ in Cancellation Configuration	54
3.4.2	Discussion	56
3.5	Summary	56
Chapter 4	Joint Time Delay Estimation and Adaptive Minimum Mean Squared Error Filtering: The Joint Least-Mean-Square Algorithm	58
4.1	Introduction	58
4.2	Convergence of the Joint LMS Algorithm Using the ODE Method	60
4.3	Analysis of the Joint LMS Algorithm in Steady-State	63
4.3.1	The Joint LMS Algorithm in Type I Configuration	65
4.3.2	The Joint LMS Algorithm in Type II Configuration: Delay in Adaptive Branch	79
4.3.3	The Joint LMS Algorithm in Type II Configuration: Delay in Reference Branch	87
4.4	Discussion	89
4.5	Application of the Joint LMS Algorithm	91
4.6	Summary	92
Chapter 5	Joint Time Delay Estimation and Adaptive Recursive Least Squares Filtering: Fast Transversal Filter Algorithms	94
5.1	Introduction	94
5.2	Background Theory	98
5.2.1	Notation and Definitions for a Type II-DRB Configuration	98
5.2.2	Notation and Definitions for a Type I Configuration	101
5.2.3	Geometrical Considerations	101
5.3	Geometrical Derivation of Lag-Recursive Relations	103
5.3.1	Derivation for a Type II-DRB Configuration	103
5.3.2	Lag Recursions for a Type I Configuration	110
5.3.3	Discussion	111
5.4	Joint Time Delay Estimation and Adaptive RLS Algorithms with the Lag-Recursive Relations in Type II-DRB Configuration	112
5.4.1	The Joint Algorithm for a Type II-DRB Configuration	114
5.4.2	Discussion	116
5.5	Analysis of the Joint LS Algorithm in Steady-State	118
5.5.1	The Joint LS Algorithm in Type II-DRB Configuration	118
5.5.2	The Joint LS Algorithm in Type I Configuration	124
5.5.3	Discussion	124
5.6	Summary	125

Chapter 6	Experimental Results: The Joint LMS Algorithm and the Joint RLS Algorithm	126
6.1	Introduction	126
6.2	Experimental Set-Up	127
6.3	Results with The Joint LMS Algorithm	129
6.3.1	Simulation of the LMS Algorithm	130
6.3.2	Multiple Convergence Points and Excess MSE	133
6.3.3	Delay Tracking Bounds	133
6.3.4	Delay Tracking Simulations in Type I	136
6.3.5	Delay Tracking Simulations in Type II	145
6.3.6	Discussion	147
6.3.7	Steady-State Results	150
6.4	Results with the Joint RLS Algorithm in Type II-DRB Configuration	156
6.4.1	The Sum of Squared Errors	157
6.4.2	Numerical Stability	158
6.4.3	Tracking Properties	164
6.4.4	Discussion	164
6.5	Results for a Reverberant Room Reference Impulse Response	164
6.5.1	Results with the Joint LMS Algorithm in Type I	168
6.5.2	Results with a Joint Hybrid LMS Delay - RLS Filter in Type II-DRB	174
6.6	Summary	182
Chapter 7	Conclusions	183
7.1	Summary	183
7.2	Contributions	187
7.3	Future Work	188
Appendix A.	Derivation of the Joint Maximum Likelihood Estimator For a Type I System	190
A.1	Derivation of The Log-Likelihood Function	190
A.2	Derivation of Entries of $Q_2(n, m d, \mathbf{w})$	193
A.3	The function $\ell_Y(d, \mathbf{w})$ for a Long Observation Interval	200
Appendix B.	The Ordinary Differential Equation (ODE) Method	203
Appendix C.	Cross-Correlation of Differentiated Random Processes	205
C.1	Cross-correlation of $\dot{x}(t)$ and $y(t)$	205
C.2	Cross-correlation of $x(t)$ and $\dot{y}(t)$	206
C.3	Cross-correlation of $\dot{x}(t)$ and $\dot{y}(t)$	206
C.4	Cross-correlation of $\ddot{x}(t)$ and $y(t)$	206
C.5	Cross-correlation of $x(t)$ and $\ddot{y}(t)$	206
Appendix D.	Some Expected Values For a Type I Adaptive System	207

D.1	Expected Value of G_n	207
D.2	Expected Value of $(1 - 2\alpha G_n)N_n$	208
D.3	Expected Value of G_n^2	209
D.4	Expected Value of N_n^2	210
Appendix E.	Shift Invariance Properties and Common Recursions in the LS algorithm: Type II-DRB	211
E.1	Shift Invariance Properties in the LS algorithm:Type II-DRB	211
E.2	Common Recursions	212
Appendix F.	Basic Fast Transversal Filter Algorithm	214
Appendix G.	Matrix-based Derivation of the Error and Weight Vector Recursions: Type II-DRB	216
G.1	Recursions for the Error	216
G.2	Recursions for the LS Weight Vector	220
G.3	Auxiliary recursions	221
<i>References</i>	225

List of Figures

2.1	Mathematical signal model	11
2.2	System models of interest; (a) Type I model, (b) Type II model	12
2.3	Blockdiagram of the noncausal joint maximum likelihood estimator (canonical realization number 1)	14
2.4	Blockdiagram of the causal joint maximum likelihood estimator (canonical realization number 1)	15
2.5	Blockdiagram of an approximate noncausal joint maximum likelihood receiver	17
2.6	System identification (cancellation) configuration	20
2.7	Inverse filtering (equalization) configuration	21
2.8	Type I systems in cancellation configuration	21
2.9	Adaptive system with sampling rate conversion	22
2.10	Type I adaptive system with sampling rate conversion	24
3.1	General model for (a) a Type I adaptive system and for (b) a Type II adaptive system	28
3.2	Type II systems, with negative delay, in cancellation configuration	30
5.1	Geometrical interpretation of (5.51)	107
5.2	Geometrical interpretation of (5.64)	107
5.3	Interpretation of the lag $\ell - 1$, ℓ and $\ell + 1$ error computations, in terms of transversal filters	113
5.4	Minimum sum of squared errors versus the continuous delay d_n , $\beta = 0.9$	119
6.1	Reference filter impulse response $h(n)$	128
6.2	Reference filter transfer function $H(e^{j\omega})$	128
6.3	Blockdiagram of the simulation of a Type I configuration	131
6.4	Reference filter deterministic autocorrelation function $\rho_h(n)$	132
6.5	The MSE function for different fixed reference delays D_n ; continuous curve: $D_n = 0.0$, large dashes curve: $D_n = 0.5$, medium dashes curve: $D_n = 1.0$, small dashes curve: $D_n = 1.5$	134
6.6	Expanded view of Fig. 6.5	134
6.7	Range of α satisfying Proposition 3.3	135
6.8	LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.1$, $\alpha = 0.5$	137
6.9	LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.1$, $\alpha = 0.1$	137
6.10	LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$	138

6.11	LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.1$	138
6.12	LMS Adaptive delay response to different reference delay step; dashed curves: reference delays; $\mu = 0.01, \alpha = 0.5$	139
6.13	Learning curve for a reference delay unit step; $\mu = 0.01$ and $\alpha = 0.5$	140
6.14	Learning curve for a reference delay unit step; $\mu = 0.01$ and $\alpha = 0.1$	140
6.15	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.5$	141
6.16	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.1$	142
6.17	Learning curve for a reference delay ramp of 0.01 sample/sample; $\mu = 0.01$ and $\alpha = 0.1$	142
6.18	LMS Adaptive delay response to a sinusoidal reference delay variation, period = 500 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.5$	143
6.19	LMS Adaptive delay response to a sinusoidal reference delay variation, period = 1000 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.5$	144
6.20	LMS Adaptive delay response to a sinusoidal reference delay variation, period = 1000 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.1$	144
6.21	LMS Adaptive delay response to a reference delay unit step in noisy conditions, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.5$	146
6.22	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample in noisy conditions, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01$ and $\alpha = 0.5$	146
6.23	LMS Adaptive delay response to a sinusoidal reference delay variation in noisy conditions, period = 1000 samples, amplitude = 1 sample, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01, \alpha = 0.5$	147
6.24	LMS Adaptive delay response to linear phase and amplitude variations in the reference filter; variations of 0.001 sample/sample; $\mu = 0.01, \alpha = 0.5$	148
6.25	Adaptive filter impulse response after 1000 iterations for the reference filter variations of Fig. 6.24	148
6.26	LMS Adaptive delay response to a reference delay unit step in Type II-DAB configuration; long dashed curve: reference delay; medium dashed curve: $\mu = 0.01, \alpha = 0.5$; continuous curve: $\mu = 0.01, \alpha = 0.1$	149

6.27	LMS Adaptive delay response to a reference delay unit step in Type II-DRB configuration; long dashed curve: reference delay; medium dashed curve: $\mu = 0.01, \alpha = 0.5$; continuous curve: $\mu = 0.01, \alpha = 0.1$	149
6.28	Theoretical curve of α versus μ for a Type I system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$	151
6.29	Theoretical curve of α versus μ for a Type II-DRB system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$	151
6.30	Theoretical curve of α_{max} versus μ for a Type I system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$	152
6.31	Theoretical curve of α_{max} versus μ for a Type II-DRB system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$	152
6.32	Theoretical curve of α versus v_{ss} for a Type I system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB	154
6.33	Theoretical curve of α versus v_{ss} for a Type II-DRB system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB	154
6.34	Theoretical curve of α_{max} versus v_{ss} for a Type I system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB	155
6.35	Theoretical curve of α_{max} versus v_{ss} for a Type II-DRB system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB	155
6.36	Measured misadjustment for a Type I system versus the steady-state delay D , SNR = 10 dB, $\mu = 0.01, \alpha = 0.5$; continuous curve: adaptive filter alone, dashed curve: joint adaptive system	156
6.37	Measured excess MSE for a Type II-DAB system versus the steady-state delay D , $\mu = 0.01, \alpha = 0.5$	157
6.38	Minimum sum of squared errors versus time, for different relative delays Δ and for $\beta = 0.97$; the lowest curve is for $\Delta = -1$, the middle curve is for $\Delta = 2$ and the upper one if for $\Delta = 6$	158
6.39	Fig. 6.38 on a vertical log scale	159
6.40	Minimum sum of squared errors versus $D - d$, $\beta = 0.97$	159
6.41	Minimum sum of squared errors versus time, for different values of β and for a relative delay of two sample; the lowest curve is for $\beta = 0.9$, the middle is for $\beta = 0.94$ and the upper one is for $\beta = 0.98$	160
6.42	Fig. 6.41 on a vertical log scale	160
6.43	Measured expected value of the minimum sum of squared errors versus β	161
6.44	Measured variance of the minimum sum of squared errors versus β	161

6.45	Behaviour of $\hat{\xi}_{M_o}(n, \ell - 1)$ with parallel restart every 500 iterations, $\beta = 0.92$.	163
6.46	Behaviour of the parallel $\hat{\xi}_{M_o}(n, \ell - 1)$ with parallel restart every 500 iterations, $\beta = 0.92$.	163
6.47	Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, noiseless conditions	165
6.48	Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 30 dB	165
6.49	Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 20 dB	166
6.50	Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, noiseless conditions	166
6.51	Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 30 dB	167
6.52	Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 20 dB	167
6.53	Impulse response of the reverberant room	168
6.54	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample and for a 200-tap reference impulse response; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$	170
6.55	LMS Adaptive delay response to a sinusoidal reference delay variation and for a 200-tap reference impulse response; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$	171
6.56	Learning curve for the joint algorithm facing a reference delay ramp of 0.01 sample/sample (corresponding to Fig. 6.54); $\mu = 0.01$, $\alpha = 0.02$	171
6.57	Learning curve for the single adaptive filter facing a reference delay ramp of 0.01 sample/sample (note the scale difference with Fig. 6.56); $\mu = 0.01$	172
6.58	Learning curve for the joint algorithm facing a sinusoidal reference delay (corresponding to Fig. 6.55); $\mu = 0.01$, $\alpha = 0.02$	172
6.59	Learning curve for the single adaptive filter facing a sinusoidal reference delay; $\mu = 0.01$ (note the factor of 10 compared to the scale of Fig. 6.58)	173
6.60	Impulse response of the adaptive filter in the joint algorithm, after 1000 iterations, when the reference delay is a ramp of 0.01 sample/sample and $\mu = 0.01$, $\alpha = 0.02$	173
6.61	Filter transfer function for coloured input generation	174
6.62	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample, for a 200-tap reference impulse response and a coloured input; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$	175

6.63	LMS Adaptive delay response to a sinusoidal reference delay variation, for a 200-tap reference impulse response and a coloured input; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$	175
6.64	Speech segment used for simulations; the dashed line indicates the range of data used for delay tracking	176
6.65	Normalized LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample, for a 200-tap reference impulse response and a speech input; dashed curve: reference delay; $\mu = 10^{-5}$, $\alpha = 1000$	177
6.66	Normalized LMS Adaptive delay response to a sinusoidal reference delay variation, for a 200-tap reference impulse response and a speech input; dashed curve: reference delay; $\mu = 10^{-5}$, $\alpha = 1000$	177
6.67	LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 0.02$	179
6.68	LMS Adaptive delay response to a sinusoidal reference delay variation when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 0.02$	179
6.69	Learning curve for the joint hybrid algorithm facing a delay ramp of 0.01 sample/sample; $\beta = 0.92$, $\alpha = 0.02$	180
6.70	Learning curve for the single adaptive filter facing a reference delay ramp of 0.01 sample/sample (note the scale difference with Fig. 6.69); $\beta = 0.92$	180
6.71	Normalized LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 2000$	181
6.72	Normalized LMS Adaptive delay response to a sinusoidal reference delay variation when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 2000$	181
A.1	Blockdiagram of the noncausal joint maximum likelihood estimator (canonical realization number 1)	193
A.2	Structure of the discrete-time noncausal linear MMSE point estimator of $s(n)$ from $\mathbf{r}(n)$, $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$, conditioned on the parameters d and \mathbf{w} , as defined in the Theorem.	195
F.1	Fast Transversal Filter Interpretation	214

List of Symbols and Abbreviations

1	vector with all elements equal to one
$\mathbf{a}_m(n)$	one-step forward linear predictor vector of order m
α	gain factor in adaptive delay algorithm
$\alpha_M(n, \ell)$	<i>a priori</i> estimation error in LS estimation
$\mathbf{A}(n i, M)$	matrix of shifted input n -vectors, defined in (5.17)
b	weight vector estimate bias
$\mathbf{b}_m(n)$	one-step backward linear predictor vector of order m
β	weighting factor in LS estimation
$b_m(n)$	backward <i>a posteriori</i> prediction error of order m
$B_m(n)$	weighted sum of the $b_m(i)$'s
B	matrix defined in (4.106)
BPSK	Binary Phase Shift Keying
\mathcal{B}	real constant between zero and one
$c_w(n)$	convolution of $w(n)$ with itself
\mathbb{C}^n	vector space of order n over the complex field
$\chi(n, D)$	steady-state noise at output of adaptive branch
$d(n)$	adaptive delay or delay estimate
$D(n)$	reference delay
\hat{D}	estimate of D
$\mathbf{D}^\ell(n)$	n -vector of reference samples, for a lag ℓ and used in LS estimation
DAB	Delay in adaptive branch (in Type II)
DRB	Delay in reference branch (in Type II)
$\delta(n)$	discrete-time impulse
$e(n), e(n, d_n)$	error signal between adaptive and reference branches
$e_M(n, \ell)$	<i>a posteriori</i> estimation error in LS estimation
$e_{\min}(n, D)$	error between optimum adaptive branch and reference branch
$E[\cdot]$	expected value operator
ξ_n	mean squared error function
$\tilde{\xi}(n)$	$E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n]$
$\hat{\xi}_M(n)$	minimum sum of weighted squared errors with respect to $\mathbf{w}(n)$ and ℓ
$\hat{\xi}_{M_0}(n, \ell)$	minimum sum of weighted squared errors with respect to $\mathbf{w}(n)$
$\xi_n^{(C)}, \xi_n^{(E)}$	mean squared error function for cancellation or equalization configuration

$\xi\{d_n, w_{dn}\}$	MSE function for specific values of the estimators
$\xi_o(d_n)$	MSE function optimized with respect to the weight vector only
ξ_{\min}	minimum of MSE function
ξ_{ex}	excess mean squared error
ξ_{ex}^d	excess mean squared error specific to the delay
ξ_{ex}^f	excess mean squared error specific to the filter
ξ_{ex}^{df}	cross-product excess mean squared error
ξ_{ss}	steady-state value of ξ_n
$\mathcal{E}(n)$	sum of exponentially weighted squared errors
$\mathbf{E}_M^{\ell}(n)$	n -vector of <i>a posteriori</i> errors, used in LS estimation
$\mathbf{E}_{M-1}^f(n)$	n -vector of forward <i>a posteriori</i> prediction errors (LS estimation)
$\mathbf{E}_{M-1}^b(n)$	n -vector of backward <i>a posteriori</i> prediction errors (LS estimation)
$\boldsymbol{\eta}_n$	noise weight vector
$\eta_m(n)$	forward <i>a priori</i> prediction error of order m (LS estimation)
f	frequency variable (Hz) (continuous-time)
$f_m(n)$	forward <i>a posteriori</i> prediction error of order m (LS estimation)
$F_m(n)$	weighted sum of the $f_m(i)$'s
$\mathbf{F}[\cdot]$	discrete-time Fourier transform operator
FIR	Finite Impulse Response
FSE	Fractionally Spaced Equalizer
FTF	Fast Transversal Filter
$g_1^{(c)}(t, v w(t))$	impulse response of the continuous causal point linear MMSE estimator of $s(t)$ in ML estimation
$\mathbf{g}_M(n)$	Kalman gain vector of order M in RLS algorithm
γ_n	estimate of MSE function
$\gamma_M(n)$	error between Kalman filter and the value one in FTF algorithm
G_n	$1/2\dot{N}_n$
$h(n)$	reference linear filter impulse response
$h^{-1}(n)$	inverse of reference linear filter impulse response
$\hat{h}, \hat{\mathbf{h}}$	estimate of $h(n)$ or \mathbf{h}
$H(e^{j\omega})$	Fourier transform of $h(n)$
\mathbf{I}	identity matrix
k	discrete-time correlation argument
$\mathbf{K}_{\eta}(n)$	noise weight vector covariance matrix
$\mathbf{K}_D(n)$	$E[\boldsymbol{\eta}_{nT-d_n}\boldsymbol{\eta}_{nT-d_n}^T]$

ℓ	integer delay estimator in LS estimation (lag)
$\ell(d, \mathbf{w})$	log-likelihood function
$\ell_Y(d, \mathbf{w})$	main term in log-likelihood function
$\ell_B(d, \mathbf{w})$	bias term in log-likelihood function
LMS	Least-Mean-Square
$\mathcal{L}_{d,g(n)}[\cdot]$	linear operator applying a delay d and a filter $g(n)$ in cascade
$\mathcal{L}_{d,g(n)}^I[\cdot]$	Type I linear operator
$\mathcal{L}_{d,g(n)}^{II}[\cdot]$	Type II linear operator
LS	Least Squares
λ	eigenvalue of autocorrelation matrix \mathbf{R}
λ_{\max}	maximum eigenvalue of \mathbf{R}
Λ	diagonal matrix with eigenvalues of \mathbf{R} on diagonal
λ	vector made of eigenvalues of \mathbf{R}
M	adaptive filter order
\mathcal{M}	misadjustment
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
μ	gain factor in adaptive filter algorithm
$n_1(n), n_2(n)$	discrete-time observation limits in ML estimation
n	discrete-time index
$N_o/2$	white noise power spectral density (W/Hz)
ν_n	delay-dependent term in MSE function
N_n	derivative noise in LMS adaptive delay algorithm
ODE	Ordinary Differential Equations method
\mathbf{p}_n	cross-correlation vector between adaptive filter input and reference signal
$\mathbf{p}(d_n)$	cross-correlation vector evaluated for a certain value d_n
$\text{Pr}[\cdot]$	probability operator
$\mathbf{P}(n)$	fluctuation matrix in analysis of RLS algorithm
$\mathbf{P}_S \mathbf{x}$	projection of a vector \mathbf{x} onto a subspace S
$\mathbf{P}_S^\perp \mathbf{x}$	orthogonal projection of a vector \mathbf{x} onto a subspace S
$\phi_{xx}(\cdot)$	autocorrelation function of signal $x(t)$ or $x(n)$,
$\phi_{ab}(\cdot)$	cross-correlation function between a signal a and a signal b depending on nature of argument

$\tilde{\Phi}_M(n)$	perturbation matrix in analysis of RLS algorithm
$\Phi_{xx}(e^{j\omega})$	power spectral density of stochastic process $x(n)$
Φ_{\max}	maximum of input power spectral density
Φ_{ss}	power spectral density of <i>white</i> input $s(n)$
$\Phi_{\mathbf{x}\mathbf{x}}(k d, \mathbf{w})$	covariance matrix of some vector of the form $\mathbf{x}(n d, \mathbf{w})$
$\Phi_M(n)$	deterministic autocorrelation matrix of adaptive branch input $u(n)$
$\psi_m(i)$	backward <i>a priori</i> prediction error of order m
\mathbf{Q}	unitary matrix with orthonormal eigenvectors of \mathbf{R} as columns
$Q_2(n, m d, \mathbf{w})$	matrix impulse response of the noncausal point linear MMSE estimator of $s(n d, \mathbf{w})$ in ML estimation
$r(n)$	reference signal
$\hat{r}(n)$	optimum adaptive branch output
\mathbf{R}, \mathbf{R}_n	$M \times M$ autocorrelation matrix of adaptive branch input $u(n)$
RLS	Recursive Least Squares
$\rho_x(k)$	deterministic autocorrelation of an impulse response $x(n)$
$\text{Re}[\cdot]$	real value operator
$s(n)$	transmitted input signal
$\mathbf{s}(n)$	M -order input vector
$\mathbf{s}(n d, \mathbf{w})$	model for the noiseless received vector, given the parameters d and \mathbf{w}
SD	Steepest-Descent
SNR	Signal-to-Noise-Ratio
$S(n i, M)$	vector subspace spanned by the columns of $\mathbf{A}(n i, M)$
$\sigma_{v_i}^2$	variance of noise v_i
t	continuous-time variable
T	sampling period
τ	continuous-time correlation argument
τ_{del}	time constant of adaptive delay algorithm
τ_j	time constant of j^{th} mode of adaptive weight vector
$(\tau_{\text{mse}})_j$	time constant of j^{th} mode of adaptive filter MSE function
Type I	refers to a system with a delay <i>in front</i> of a linear filter
Type II	refers to a system with a delay <i>after</i> a linear filter
$\tilde{\theta}_M^\ell(n)$	cross-correlation of input signal and reference signal for a lag ℓ
Θ_n	delay value near or equal to a minimum of the MSE function
$\theta_M^\ell(n)$	deterministic cross-correlation vector between adaptive filter input

	and reference signal for lag ℓ
$u(n)$	adaptive branch input signal
\mathbf{u}_n	equivalent vector of delayed input samples
$\mathbf{u}_M(n)$	M^{th} -order equivalent vector of delayed input samples
U_1, U_2	inflexion points on each side of global minimum of MSE function
$\mathbf{U}(n)$	n -vector of adaptive branch input samples, used in LS estimation
$v_1(n), v_2(n)$	additive noise processes
v_n	delay estimate variance
v_{ss}	steady-state delay estimate variance
$v_{M-1}^{f\ell}(n)$	complex conjugate of inner product of forward error prediction and desired response (LS estimation)
$v_{M-1}^{b\ell}(n)$	complex conjugate of inner product of backward error prediction and desired response (LS estimation)
$\mathbf{v}(n)$	2-dimensional vector with components $v_1(n)$ and $v_2(n)$
$w(n)$	impulse response estimator in ML estimation
$\mathbf{w}, \mathbf{w}_n, \mathbf{w}(n)$	weight vector estimator
$\mathbf{w}_M^\ell(n)$	weight vector estimator for a given lag ℓ in LS estimation
w_{ni}	i^{th} component of \mathbf{w}_n
\mathbf{w}_{dn}	delayed version of weight vector
\mathbf{w}_{opt}	MMSE weight vector
$\hat{\mathbf{w}}_M^\ell(n)$	weight vector that minimizes $\mathcal{E}(n)$ for a given ℓ in LS estimation
w.r.t.	with respect to
ω	frequency variable (radians) (discrete-time)
$\mathbf{X}(n)$	transformation of $\mathbf{K}_\eta(n)$ with \mathbf{Q}
\mathbf{X}_{ss}	steady-state value of $\mathbf{X}(n)$
$\mathbf{x}(n)$	vector made of diagonal elements of $\mathbf{X}(n)$
\mathbf{x}_{ss}	steady-state value of $\mathbf{x}(n)$
$y(n)$	output of adaptive branch
$y_1(n), y_2(n)$	received signals available for joint estimation
$\mathbf{y}(n)$	2-dimensional vector with components $y_1(n)$ and $y_2(n)$
z^{-j}	vector shift operator, defined in (5.16)
$z_1(n), z_2(n)$	transformed noise sources
\otimes	convolution operator
\oplus	direct sum operator
$(\cdot)^*$	complex conjugate operator

∇	gradient operator
$\hat{\nabla}$	estimated gradient
$(\cdot)^T$	transpose operator
$(\cdot)^H$	complex conjugate transpose
$\dot{x}(d_n)$	first derivative of a function $x(\cdot)$, with respect to the delay
$\ddot{x}(d_n)$	second derivative of a function $x(\cdot)$ with respect to the delay
$x'(\cdot)$	first derivative of a function $x(\cdot)$ with respect to the correlation or time argument
$x''(\cdot)$	second derivative of a function $x(\cdot)$ with respect to the correlation or time argument
$x^{(i)}(\cdot)$	i^{th} derivative of a function $x(\cdot)$ with respect to the correlation or time argument
$x^{[i]}(d_n)$	i^{th} derivative of a function $x(\cdot)$ with respect to the delay
\propto	proportional to
\propto^{-1}	inversely proportional to
$\text{tr}[\cdot]$	trace operator
$\langle \cdot \rangle$	time average operator
$\ \mathbf{x}\ $	norm of a vector \mathbf{x}
$\langle \mathbf{x}, \mathbf{y} \rangle$	inner product operator between vectors \mathbf{x} and \mathbf{y}
$\{\mathbf{x}\}$	vector space spanned by \mathbf{x}
$S \cup \{\mathbf{x}\}$	vector space spanned by a subspace S and a vector \mathbf{x} not in S
$[\mathbf{x}]_m$	vector made of the m top components of vector \mathbf{x}
$[\mathbf{x}]_m$	vector made of the m bottom components of vector \mathbf{x}

1.1 Conventional Adaptive Filtering versus Delay Estimation

Adaptive digital signal processing has become an important part of many systems involving unknown components or nonstationary subsystems. Adaptive digital filters, under different forms, are commonly used in channel equalization [1], echo cancellation [2], noise cancellation [3], system identification [4], spectral analysis [5] and in many other signal processing tasks [6]. Much research related to adaptive filters is concerned with the convergence, the tracking and the computational complexity of the adaptive algorithms [7]. It is almost always assumed that the two main digital inputs to the algorithm, the adaptive filter input signal and the reference signal, are synchronized in time, i.e. that they are the sampled versions of two continuous signals, with the sampling clock being the same for both.

But in some adaptive filtering applications, this assumption is not true. A sampling rate difference makes the input and reference signals jointly nonstationary, and the two sequences used in the adaptive filter experience a changing relative delay. The reference system, if it is linear, can then be modelled as a reference linear filter in series with a **time-varying delay**. This delay decorrelates the two signals as the time index increases. In some other forms of adaptive system modelling, the unknown system has an impulse response that can be explicitly modelled as a pure time delay in series with a linear filter. Examples of such systems occur in geophysical exploration [8], echo cancelling [9] or multipath communications [1].

The ability of an adaptive filter, operating at or above the signals Nyquist rate, to model a delay between the filter primary input and its reference, makes it a very versatile signal processing tool and, in many cases, the designer does not need to consider any other

delay compensation scheme. The adaptive filter essentially models the delay by shifting its impulse response by the proper amount. The use of a simple adaptive filter, to identify the reference system, implies that the combination of the delay and the filter will be modelled by the adaptive system, without any explicit separation between the delay and the filter estimates. In some cases this is sufficient, but it can also happen that the estimate of interest is the delay value, as in *delay estimation over an unknown channel*, or that the channel impulse response is wanted, as in *channel identification with an unknown delay* (these different interpretations are indeed very similar and are most often related to the perspective of the user).

Even if a separation between the delay estimate and the channel estimate is not required, a simple adaptive filter might require a number of filter weights, of which many may have no effect upon the final model (because they are used only to delay the input signal), but increase both the computational complexity and the weight vector misadjustment, resulting in an increased mean squared error. For a given misadjustment, such a large number of weights has usually the effect of reducing the convergence speed of the adaptive filter and its tracking capability [10]. In the case of a time-varying reference delay, the sampling rate evolution can even be rapid enough to prevent the tracking by a conventional adaptive filter [11], [12]. For some applications, it is therefore imperative to have some appropriate means to "center" the impulse response of the adaptive filter within a finite time window.

The separation of the estimation task, between a delay estimator and a linear filter estimator, has been given very little attention in the adaptive filtering literature. The exception is in the field of clock or timing recovery used in conjunction with adaptive equalizers, in data communication systems [13]. In digital channel equalization, for example, the receiver input signal (or a filtered version of it) is sampled and passed through an adaptive filter (the equalizer). The reference signal is the demodulated data stream or a locally remodulated version of it. Due to channel delay distortion or some other nonstationary channel effects, the sampling phase has to be synchronized with the locally generated reference signal. Some form of equalization strategies will compensate for this sampling error, as in fractionally spaced equalizers (FSE), by adjusting their taps to model the corresponding delay [14]. But this scenario explicitly assumes that the sampling period has been recovered, and that only the clock phase has to be tracked (this implicitly means that a form of carrier phase recovery is performed independently from the equalizer).

There are other applications in which the difference in sampling rates, between the adaptive filter input and its reference signal is implicit. A particular example of such an application is the enhancement of speech in the presence of interfering music and noise [12].

An adaptive noise canceller is used to model the channel through which the speech and the interference are transmitted, and its output is subtracted from the composite signal, in order to obtain the enhanced speech. But, due to different recording media, there is a difference in sampling rates between the discrete composite signal and the interference signal. The signal decorrelation caused by this difference renders the noise canceller useless after a few seconds of operation, and methods to “realign” the canceller input and its reference signal are essential.

The study of such methods is the subject of this thesis. Since time delay estimation is an inherent part of the algorithms considered in the next chapters, conventional methods to perform such a task are reviewed in the next section. Section 1.3 addresses briefly the subject of conventional adaptive filtering. The main thesis objectives are given in Section 1.4, where the estimator structure that is favoured all along the work is introduced. Joint estimation algorithms involving adaptive filters are discussed in Section 1.5.

1.2 Conventional Delay Estimation

The signal model, virtually always assumed in the delay estimation literature, is composed of two received noisy signals, one being a delayed and scaled version of the other, with additive noise processes uncorrelated with each other. As in most estimation problems, both open-loop and closed-loop methods have been proposed for time delay estimation. Most of these methods make use, either explicitly or implicitly, of the cross-correlation between the received signals or a filtered version of them. In the *generalized correlation method*, as discussed by Knapp and Carter [15], the two received signals are first filtered by different filters, and one output is delayed with respect to the other. The resulting signals are correlated together, for different values of delays, until a maximum in the cross-correlation is obtained. This configuration is used with different filter combinations, each one emphasizing a different characteristic of the signals. Assuming that all the signals are stationary and Gaussian, Knapp and Carter derive the filters giving the maximum likelihood (ML) open-loop delay estimator for a constant delay. These results are generalized, for time-varying delays, by Stuller [16] and by Champagne et al. [17].

For time-varying delays, Meyr and Spies [18] propose the use of the ML estimator in a closed-loop configuration. Using a small error signal assumption, the system is analyzed by converting it into a mathematically equivalent *delay-locked loop*, bearing a great resemblance to the conventional phase-locked loop. The delay-locked loop is composed of a delay error generator, an integrator and a loop filter. Messer [19] analyzes the same type of closed-loop

configuration for different kinds of delay error generator, all based on the cross-correlation approach.

Closed-loop adaptive techniques using the minimum mean squared error (MMSE) or the least squares (LS) criteria have been proposed by many authors. The basic configuration adopted by these researchers is the system identification one. In this structure, one signal is processed by an adaptive system and the output is compared to the other signal, in order to produce the error signal for adaptation. The conventional adaptive transversal filter was proposed for the modelling of the delay and attenuation experienced by the reference signal. This method relies on the fact that a pure time delay can be imposed on a band-limited continuous lowpass signal by passing this signal through a lowpass filter with a frequency response constant in amplitude and linear in phase [20]. This frequency response corresponds to a $\sin x/x$ impulse response and can be approximated by a digital finite impulse filter (FIR) of appropriate length. The least-mean-square (LMS) algorithm has been studied by Reed et al. [21] and Krolik et al. [22] for static delays and by Feintuch et al. [23] as well as by Youn and Carter [24] for time-varying delays. Chan et al. [25], [26] have considered the RLS algorithm. In these methods, the adaptive filter converges to the Wiener solution and a subsequent interpolation algorithm determines the delay estimate as the peak location of the adapted impulse response. This delay estimator is biased because of the finite interpolation process between the adaptive filter coefficients [27]. Note that in these methods, the adaptive filter converges to a solution that is a function of the input signal autocorrelation.

In the above adaptive method, the adaptive filter identifies the channel impulse response (the $\sin x/x$ function) and the delay estimate is obtained by measuring the displacement of this response. Therefore, in order to estimate a scalar parameter, the whole weight vector must be estimated and processed. Adaptive approaches, in which the delay value is directly estimated, have also been proposed. These use the basic identification configuration described above, with the exception that the conventional adaptive transversal filter is replaced by an adaptive delay element. The delay is adapted directly, until the MMSE or the LS solution is reached. The LMS delay adaptation algorithm has been studied by Etter and Stearns [28], for integer delay values, and by Messer and Bar-Ness [29], for fractional delay values.

Instead of the MMSE criterion, Smith and Friedlander [30] consider the *weighted* LS criterion and the Gauss-Newton adaptation method for a fractional delay element. They claim that the method is better suited than the LMS algorithm for time-varying delay tracking.

The delay estimations methods based on the use of a delay element are conceptually simpler than those based on the adaptive transversal filter, but they show one major drawback; the algorithm is not guaranteed to converge to the delay corresponding to the global minimum of the performance surface, since this surface is not in general unimodal with respect to the adaptive delay value (it depends on the input signal autocorrelation function). This implies that in the case of a cold start of the algorithm, an acquisition procedure is necessary to bring the delay value in the vicinity of the global minimum and allow this minimum to be tracked by the algorithm.

1.3 Conventional Adaptive Transversal Filtering

Traditionally, the subject of adaptive transversal filtering has been divided into two subclasses, referring to the two most popular estimation criteria used in the adaptation algorithm [7]. The gradient-based algorithms (steepest-descent and LMS), make use of the MMSE criterion, while the recursive least squares (RLS) algorithm is based on the LS criterion.

The steepest-descent algorithm is based on the conventional nonlinear programming method bearing the same name [31]. In this method, the adaptive weight vector is updated using a scaled version of the gradient of the mean squared error function, with respect to the weight vector. The MSE function is defined as the expected value of the squared error between the filter output and the reference signal. This function is quadratic with respect to the weight vector, and its gradient is linear. The computation of the gradient requires the input signal autocorrelation matrix, as well as the cross-correlation vector between this input and the reference signal. In practice, these values have to be estimated if the SD is to be applied. The LMS algorithm is an attempt to simplify the gradient estimation, in which it is assumed that the MSE function is replaced by the squared error function. This gives a gradient vector estimate that is equal to minus twice the input signal vector multiplied by the error, which reduces considerably the algorithm's complexity. The LMS adaptation algorithm is therefore a stochastic gradient algorithm that is simple and reliable, and that has been used in many adaptive signal applications. A major problem related to the SD algorithm is its slow convergence properties, which are related to the magnitude of the smallest eigenvalue of the input signal autocorrelation matrix, as well as to the eigenvalue spread [6]. Methods to speed up the convergence have been proposed. In these methods, a form of whitening of the input signal is performed or used, in order to lower the eigenvalue spread.

The above gradient-based methods are implemented, most of the time, in the time domain, although various frequency-domain methods have been proposed [32]. The advantage of this structure is that, for filters with a large number of coefficients, the use of fast Fourier transforms to convert the different signals in the frequency domain (where the adaptation and the filtering are accomplished) reduces dramatically the computational complexity of the algorithm.

The LS-based estimation has for goal the minimization of the (weighted) sum of squared error over a window of increasing length. The weight is selected to be less or equal to one, which practically limits the memory of the algorithm and allows the tracking of nonstationary systems and signals. The computation of the LS solution essentially involves the inverse of the deterministic input signal autocorrelation matrix, which is obtained under a form of time average. This inverse can be computed recursively in time, and gives rise to the recursive LS (RLS) algorithm. Because this algorithm makes use of the matrix inverse at each iteration, which is equivalent to an input whitening, its convergence rate is typically an order of magnitude larger than that of the LMS algorithm [7]. The RLS algorithm is computationally involved and different forms of “fast” algorithms have been proposed. The drawback of these efficient methods is their inherent computational instability on finite word length processors.

As far as tracking possibilities are concerned, the RLS algorithm, although it converges faster, does not seem to be superior to the LMS algorithm for filters of low order [33], [34].

1.4 Thesis Objectives

The main objective of the thesis is to obtain and analyze some adaptive structures that would allow one to estimate separately the delay and the channel that link together two observed signals. Since conventional adaptive filter theory is fairly well understood and since its application gives good practical results, the new adaptive structures retain as much as possible the forms of the well known adaptive systems. In particular, the conventional estimation criteria, the minimum mean squared error criterion and the least squares criterion, are the main concerns of this thesis. In addition, the steepest-descent, least-mean-square and recursive least squares adaptation algorithms constitute the core of the work, as in traditional adaptive filtering theory [7].

These *joint time delay and adaptive filtering algorithms* are composed of an adaptive delay element [29] operating in conjunction with a conventional adaptive transversal filter. The delay element is essentially a delay line (implementing the integer part of the delay)

in series with an interpolation filter [35] (implementing the fractional part of the delay by resampling the input signal). These new adaptive structures meet two fundamental objectives: first, the structure of the investigated joint estimation algorithms, although simple and seemingly *ad hoc*, follows a pattern that suggests itself in a rigorous derivation of the maximum likelihood joint estimator (see Chapter 2); second, the joint MMSE or LS estimators extend the capabilities of existing adaptive delay estimators or adaptive transversal filters.

Hence, the analysis of joint algorithms, as presented in this thesis, has as an objective the extension of the existing adaptive filtering and/or adaptive delay estimation theories. It is desired to derive the critical system parameters that govern both the convergence conditions and the steady-state performance of each of the joint algorithms. This theoretical objective motivates much of the research. Practical considerations, under the form of simulations, are also provided and discussed.

1.5 A Survey of Joint Algorithms Involving Adaptive Filters

Most of the work dealing with joint algorithms and involving a form of adaptive transversal filter was performed in the field of digital communications, where the adaptive filter considered is a channel equalizer. Kobayashi [36] looks at the problem of deriving *simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems*. He seeks a joint estimator for the carrier phase, the bit timing and the symbol recovery for different forms of modulated signals. He considers the joint maximum likelihood estimator for which he defines a steepest-descent algorithm that searches the ML performance function.

Chang [37] considers the *joint optimization of automatic equalization and carrier acquisition* for BPSK signals, using the MMSE criterion and a joint steepest-descent algorithm. He studies the location and magnitude of the stationary points of the MSE function and finds that there is no local minimum or maximum and an infinitude of global minima, located π radians apart. He also derives necessary convergence conditions for the joint algorithm. Falconer addresses the same problem, for two-dimensional-modulated suppressed-carrier data signals, proposing the *joint LMS carrier phase recovery and adaptive equalization algorithm* [38], [39]. The algorithm is studied in order to establish the convergence bounds, as well as the response to different carrier phase excitations.

Qureshi studies a *joint timing recovery and adaptive equalization* algorithm in [13], for partial-response systems. He proposes a joint LMS algorithm and discusses its practical

implementation. Previously, he had considered a gradient-directed search of the error produced at the output of the adaptive equalizer, in order to find the optimum position of the reference tap [40].

The different forms of adaptive equalizers-based joint algorithms presented above represent the basic knowledge in the field and are expanded upon in this thesis. In particular, Qureshi's work is generalized in Chapter 3 and 4 (see also [41]). The convergence conditions and bounds are considered in details for general joint delay estimation and adaptive filtering algorithms.

Recursive least squares adaptation algorithms, for the same kind of general joint adaptive system, are proposed in Chapter 5 and are also discussed in [42].

1.6 Thesis Organization

The thesis is organized as follows. The subject of the next chapter is the structure of joint time delay estimation and adaptive filtering algorithms. The problem of estimating the time delay and the correlation function between two received signals is introduced in this chapter. A mathematical model is initially discussed, and a possible form for the joint maximum likelihood estimator, for the time delay and the correlation function between two observed Gaussian signals, is presented. The joint MMSE and LS algorithms, as studied in the subsequent chapters, are then introduced. The objective of this brief theoretical chapter is twofold. First of all, the structure and interpretation of an optimum (in the maximum likelihood sense) processor, as derived in Appendix A, is discussed. This represents by itself an interesting exercise in estimation theory and the general results are new. The second objective of Chapter 2 is to highlight the motivation for simpler and more practical joint estimator structure, as studied in the subsequent chapters.

Joint, gradient-based, MMSE time delay estimation and adaptive filtering algorithms are studied in Chapters 3 and 4. The MMSE theory, for joint estimation, is reviewed in Chapter 3 as a function of the different variants of the joint adaptive structure, and the joint steepest-descent algorithm is studied. In this algorithm, the derivative with respect to the delay and the gradient with respect to the weight vector are computed exactly. The convergence of the joint steepest-descent algorithm, from an arbitrary point, is studied. Then, the delay tracking properties are investigated, in general terms, and as functions of the system parameters.

The joint LMS algorithm, in which both the adaptive delay element and the adaptive filter are adapted using a stochastic gradient approximation, is studied in Chapter 4. The

convergence, from arbitrary initial conditions, is considered again, followed by an analysis of the conditions of convergence, in the mean and in the mean square, of both the estimates. The excess MSE and the misadjustment expressions resulting from the stochastic gradient approximation are derived for different variants of the joint adaptive structure.

The subject of Chapter 5 is the application of the recursive least squares algorithm (RLS) in the adaptation of the joint adaptive structure. A new form of RLS algorithm, in which the adaptive filter is adapted recursively, both in time and in the optimum delay direction, is derived. This chapter has a structure that is slightly different than the structure of Chapter 4, since it is mainly oriented toward the derivation of the joint LS algorithm, which is much more complicated than the joint SD or LMS algorithms. The excess MSE and misadjustment, caused by the finite memory of the algorithm, are also computed.

Following these theoretical chapters, Chapter 6 is more practically oriented. It presents and discusses the implementation of the joint LMS and LS algorithms and present numerous simulation results. The goal of the chapter is to confirm the applicability of the joint algorithms in different situations, and to verify the different theoretical results of the previous chapters.

Finally, Chapter 7 summarizes the thesis, discusses the contributions and gives some future research avenues.

Chapter 2

Joint Time Delay Estimation and Channel Identification

2.1 Introduction

The problem of estimating the time delay and the correlation function between two received signals is presented in this chapter. A mathematical model for the two signals is introduced. A form for the joint maximum likelihood estimator, for the delay and the correlation function, is derived, assuming Gaussian signals. Next, joint delay estimation and adaptive filtering algorithms, as studied in the subsequent chapters, are discussed. The goal of this chapter is to present the joint estimation problem in mathematical terms and to discuss the relative merits of the estimation algorithms based on different criteria.

2.2 The Mathematical Model

Two discrete signals, $y_1(n)$ and $y_2(n)$, are assumed to be available to the joint estimation algorithm. The mathematical model for the generation of these signals is

$$\begin{aligned}y_1(n) &= s(n) + v_1(n) \\y_2(n) &= \mathcal{L}_{D_n, h(n)}[s(n)] + v_2(n),\end{aligned}\tag{2.1}$$

where $s(n)$ is the transmitted stationary[†] signal and D_n is a delay, possibly time-varying. In addition, $\mathcal{L}_{D_n, h(n)}(\cdot)$ is an *unknown* linear operator, taking the form of a filtering operation, with the filter impulse response $h(n)$, of a delayed by D_n version of the input signal. The signals $v_1(n)$ and $v_2(n)$ are zero-mean stationary noise processes, assumed uncorrelated with

[†] Unless otherwise stated, stationarity means stationarity in the wide sense.

each other, as well as with $s(n)$ and $\mathcal{L}_{D_n, h(n)}[s(n)]$. A block diagram corresponding to the mathematical model (2.1) is illustrated in Fig. 2.1. Note that all the discrete signals defined above are assumed to be the sampled versions, with sampling period T , of continuous-time signals that are strictly bandlimited to the frequency range $-1/2T < f < 1/2T$.

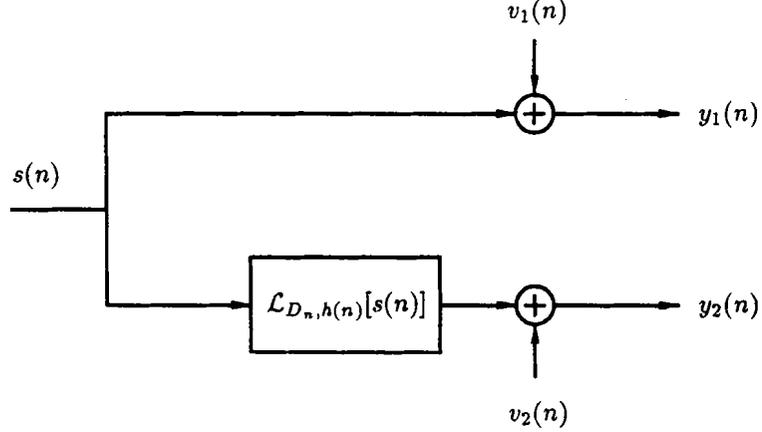


Fig. 2.1 Mathematical signal model

It is assumed that $\mathcal{L}_{D_n, h(n)}[s(n)]$ can take the two following forms:

$$\mathcal{L}_{D_n, h(n)}^I[s(n)] = h(n) \otimes s(nT - D_n), \quad (2.2)$$

corresponding to the filtering of a delayed version of $s(n)$ or

$$\mathcal{L}_{D_n, h(n)}^{II}[s(n)] = h(t) \otimes s(t)|_{t=nT-D_n}, \quad (2.3)$$

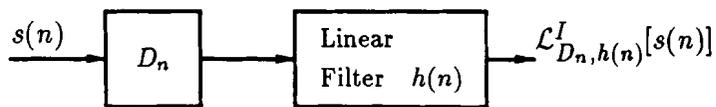
corresponding to a filter followed by a delay. Note that the operator \otimes is the convolution operator. The form of (2.2) is defined as a Type I system and the form of (2.3) as a Type II system. Note that because $h(n)$ and $s(n)$ are the sampled version of $h(t)$ and $s(t)$, $\mathcal{L}_{D_n, h(n)}^{II}[s(n)]$ is also given by

$$\mathcal{L}_{D_n, h(n)}^{II}[s(n)] = \sum_k [h(k) \otimes s(k)] \frac{\sin[\pi(t - kT)/T]}{\pi(t - kT)/T} \Big|_{t=nT-D_n}. \quad (2.4)$$

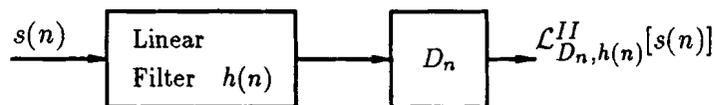
The Type I and Type II system models can be represented by the block-diagrams of Fig. 2.2. In the joint estimation problem, it is required that both the time-varying delay D_n and the reference filter $h(n)$, or its inverse $h^{-1}(n)$, be estimated [†].

The mathematical model presented in this section will be used, in Sections 2.3 and 2.4, to derive the structures of joint estimators based on the maximum likelihood criterion [43] and on the minimum mean squared error and least squares criteria [7].

[†] Note that the inverse of any linear filtering operation $h(n)$ is denoted as $h^{-1}(n)$. Therefore $h(n) \otimes h^{-1}(n) = \delta(n)$.



(a)



(b)

Fig. 2.2 System models of interest; (a) Type I model, (b) Type II model

2.3 The Joint Maximum Likelihood Estimator For a Type I System

The ML estimator has been derived by a few authors, for the identification of a pure delay between two Gaussian signals [15], [16], [17]. New results, concerning the generalization of the pioneering work appearing in these articles, are presented in this section. The derivation of these results, mainly concentrated in Appendix A, is accomplished by using basic tools in estimation theory [43]. The resulting form of the joint ML estimator provides the motivation for simpler and more practical joint estimator structures, as presented in Section 2.4 and studied in the subsequent chapters. The ML estimator for a finite observation time is presented in the next subsection and its extension for long (infinite) observation interval is discussed in Subsection 2.3.2.

2.3.1 The Joint ML Estimator for Finite Observation Interval

The parameter estimation model of (2.1) is utilized with $\mathcal{L}_{D_n, h(n)}[s(n)]$ given in (2.2). The signal $s(n)$ is assumed to be the sampled version of a continuous-time sample function $s(t)$, from a stationary zero-mean Gaussian random process with an autocorrelation function defined as $\phi_{ss}(\tau)$. The discrete-time noise processes $v_1(n)$ and $v_2(n)$ are sampled version of zero-mean stationary continuous-time Gaussian noise processes, assumed white with power spectral density $N_o/2$ W/Hz. Hence, the discrete-time noise processes have the following autocorrelation functions

$$\phi_{v_1 v_1}(k) = \phi_{v_2 v_2}(k) = \frac{N_o}{2} \delta(k), \quad (2.5)$$

with $\delta(k)$ defined as

$$\delta(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (2.6)$$

For the analysis, the reference delay D_n and the reference filter $h(n)$ are also assumed to be constant with time. Note that the assumption of equal noise variances, although seemingly artificial, is a common one in the delay estimation literature. Furthermore, in the case of the derivation of the ML receiver, it simplifies considerably the computations.

The objective is to derive an estimator producing the estimates of D and $h(n)$, defined respectively as d and $w(n)$, that maximizes the likelihood probability of the observed signals $y_1(n)$ and $y_2(n)$, over a certain discrete-time interval $[n_1, n_2]$. In order to perform this task, the mathematical model of (2.1), given some values d and $w(n)$, is expressed in the following vector form

$$\mathbf{y}(n) = \mathbf{s}(n|d, \mathbf{w}) + \mathbf{v}(n), \quad (2.7)$$

where the vectors are defined as

$$\mathbf{y}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} \quad (2.8)$$

$$\mathbf{s}(n|d, \mathbf{w}) = \begin{bmatrix} s(n) \\ \mathcal{L}_{d, w(n)}^I[s(n)] \end{bmatrix} \quad (2.9)$$

$$\mathbf{v}(n) = \begin{bmatrix} v_1(n) \\ v_2(n) \end{bmatrix}. \quad (2.10)$$

The vector \mathbf{w} is defined as the assumed reference filter weight vector, whose components are the samples of the impulse response $w(n)$. The ML estimation problem is therefore the same as computing and maximizing the likelihood probability of the received vector $\mathbf{y}(n)$, given the parameters d and \mathbf{w} , over an interval $[n_1, n_2]$. Since all signals are Gaussian, this is equivalent to the computation of a log-likelihood function $\ell(d, \mathbf{w})$. The derivation of this likelihood function is given in Section A.1 of Appendix A, using a vector form of the Karhunen-Loève decomposition [43]. The final form of this function is found to be the sum of a noncausal term $\ell_Y(d, \mathbf{w})$ and a bias term $\ell_B(d, \mathbf{w})$. Therefore,

$$\ell(d, \mathbf{w}) = \ell_Y(d, \mathbf{w}) + \ell_B(d, \mathbf{w}), \quad (2.11)$$

where

$$\ell_Y(d, \mathbf{w}) = 1/N_o \sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{y}^H(n) Q_2(n, m|d, \mathbf{w}) \mathbf{y}(m) \quad (2.12)$$

and

$$\ell_B(d, \mathbf{w}) = -\frac{1}{2} \sum_{i=1}^{\infty} \ln \left[\frac{2\lambda_i(d, \mathbf{w})}{N_o} + 1 \right]. \quad (2.13)$$

In (2.12), $Q_2(n, m|d, \mathbf{w})$ is the matrix impulse response of the noncausal linear MMSE point estimator of $\mathbf{s}(n|d, \mathbf{w})$, from the received vector $\mathbf{y}(n)$, given the parameters d and \mathbf{w} [43]. It is given by the solution of the “normal” equation

$$\frac{N_o}{2} Q_2(n, m|d, \mathbf{w}) + \sum_{k=n_1}^{n_2} Q_2(n, k|d, \mathbf{w}) \Phi_{\text{ss}}(k - m|d, \mathbf{w}) = \Phi_{\text{ss}}(n - m|d, \mathbf{w}), \quad (2.14)$$

for $n_1 \leq n \leq n_2, n_1 \leq m \leq n_2$. The matrix $\Phi_{\text{ss}}(k|d, \mathbf{w})$ is the covariance matrix of the vector $\mathbf{s}(n|d, \mathbf{w})$, defined as $\mathbf{s}(n|d, \mathbf{w})$ is zero-mean)

$$\Phi_{\text{ss}}(k|d, \mathbf{w}) = E[\mathbf{s}(n+k|d, \mathbf{w}) \mathbf{s}^H(n|d, \mathbf{w})], \quad (2.15)$$

where H denotes complex conjugate transpose. In (2.13), $\lambda_i(d, \mathbf{w})$ is the i^{th} eigenvalue of $\Phi_{\text{ss}}(k|d, \mathbf{w})$.

The form of the joint ML estimator, based on the above definitions, is given in Fig. 2.3. It is a noncausal processor, and a causal estimator can be obtained by delaying the matrix impulse response and the input vector by a value equal to the estimation interval $N = n_2 - n_1 + 1$, as shown in Fig. 2.4. The response $Q_2(n - N, m|d, \mathbf{w})$ is defined over $n_1 + N \leq n \leq n_2 + N, n_1 \leq m \leq n_2$ and the bias term has to be delayed accordingly. Note that the form of Figs. 2.3 and 2.4 is only one possible realization of the ML estimator and that other structures are possible [43]. The form of Figs. 2.3 and 2.4 is similar to the canonical realization number 1 of [43] and [16].

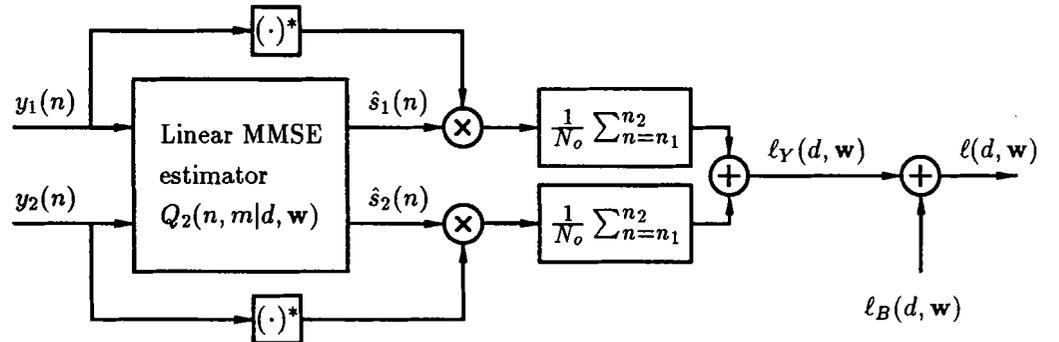


Fig. 2.3 Blockdiagram of the noncausal joint maximum likelihood estimator (canonical realization number 1)

The computation of the likelihood function can be expressed in a more appealing form by assuming that the observation time is long compared to length of the impulse response of the receiver. This is done in the next subsection.

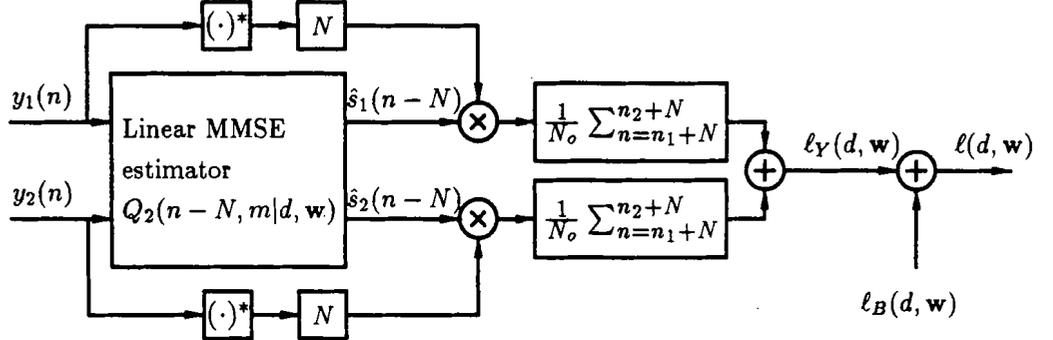


Fig. 2.4 Blockdiagram of the causal joint maximum likelihood estimator (canonical realization number 1)

2.3.2 The Joint ML Estimator for Long Observation Interval

The assumption of long (infinite) observation interval simplifies the computation of the likelihood function $\ell(d, \mathbf{w})$ by allowing the use of time-invariant filters and frequency domain relationships. This assumption is of practical importance because if the observation time is long compared with the time necessary for the system transients to die out, the estimator performs close to optimum [44]. The assumption of infinite interval is only used to solve the integral equations of the form of (2.14). The resulting receivers are still used over the interval $[n_1, n_2]$ †.

2.3.2.1 The Function $\ell_Y(d, \mathbf{w})$ for Long Observation Interval

Assume that $n_1 \rightarrow -\infty$ and $n_2 \rightarrow \infty$. Then (2.14) becomes

$$\sum_{k=-\infty}^{\infty} Q_2(k|d, \mathbf{w}) \Phi_{\mathbf{y}\mathbf{y}}(n-k|d, \mathbf{w}) = \Phi_{\mathbf{s}\mathbf{s}}(n|d, \mathbf{w}), \quad (2.16)$$

where $\Phi_{\mathbf{y}\mathbf{y}}(k|d, \mathbf{w})$ is defined as

$$\begin{aligned} \Phi_{\mathbf{y}\mathbf{y}}(k|d, \mathbf{w}) &= E[\mathbf{y}(n+k)\mathbf{y}^H(n)|d, \mathbf{w}] \\ &= E[\mathbf{s}(n+k|d, \mathbf{w})\mathbf{s}^H(n|d, \mathbf{w})] + E[\mathbf{v}(n+k)\mathbf{v}^H(n)] \\ &= \Phi_{\mathbf{s}\mathbf{s}}(k|d, \mathbf{w}) + \frac{N_o}{2}\mathbf{I}\delta(k), \end{aligned} \quad (2.17)$$

† Note that Champagne et al. [17] use a dimensionality reduction technique that eases the solution of the integral equation, in the case of pure time-delay estimation, and leads to a signal processor form that computes exactly the ML pure time-delay estimator over an arbitrary observation interval.

with \mathbf{I} representing the 2×2 identity matrix.

Taking the Fourier transform and solving, the frequency domain solution is the matrix transfer function given by

$$Q_2(e^{j\omega}|d, \mathbf{w}) = \Phi_{ss}(e^{j\omega}|d, \mathbf{w}) \Phi_{yy}^{-1}(e^{j\omega}|d, \mathbf{w}). \quad (2.18)$$

Solving the above equation and using the result in (2.12) gives, after some manipulations (see Section A.3 in Appendix A)

$$\begin{aligned} \ell_Y(d, \mathbf{w}) = & 1/2N_o \sum_n [\tilde{w}(-n|\mathbf{w}) \otimes y_1^*(nT - d)] y_2(n) \\ & + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes c_w(n) \otimes y_1(nT - d)] y_2^*(n) \\ & + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes w(n) \otimes y_1(n)] y_1^*(n) \\ & + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes w(n) \otimes y_2(n)] y_2^*(n), \end{aligned} \quad (2.19)$$

where

$$\tilde{w}(n|\mathbf{w}) = \mathbf{F}^{-1} \left[\frac{2G(e^{j\omega})W^*(e^{j\omega})}{|W(e^{j\omega})|^2 + 1} \right] \quad (2.20)$$

$$G(e^{j\omega}) = \frac{\Phi_{ss}(e^{j\omega})(|W(e^{j\omega})|^2 + 1)}{\Phi_{ss}(e^{j\omega})(|W(e^{j\omega})|^2 + 1) + N_o/2} \quad (2.21)$$

$$c_w(n) = w(n) \otimes w(n) \quad (2.22)$$

$$\Phi_{ss}(e^{j\omega}) = \mathbf{F}[\phi_{ss}(n)] \quad (2.23)$$

$$W(e^{j\omega}) = \mathbf{F}[w(n)] \quad (2.24)$$

and $\mathbf{F}[\cdot]$ is the Fourier transform operator.

2.3.2.2 Approximate Joint Maximum Likelihood Receivers

A possible realization of the receiver, based on (2.19) is illustrated in Fig. 2.5. This receiver is suboptimal, but the approximation becomes better when the observation interval increases.

The open-loop estimator operates as follows: for each possible value of d and $w(n)$ in a predetermined range of values, the likelihood $\ell(d, \mathbf{w})$ is computed over the interval $[n_1, n_2]$, using the processor of Fig. 2.5. The estimate $(\hat{D}, \hat{\mathbf{h}})$ is the pair corresponding to the likelihood maximum, over the range of values considered. In open-loop operation, the estimator is therefore conceptually made of a number (possibly infinite) of receivers operating in parallel. Every one of these parallel receivers effectively computes the likelihood of a certain couple. By quantizing the range of possible solutions, the number of receivers is reduced from an infinity to a finite number (although very large in the case of a multicomponent vector \mathbf{w}) [43].

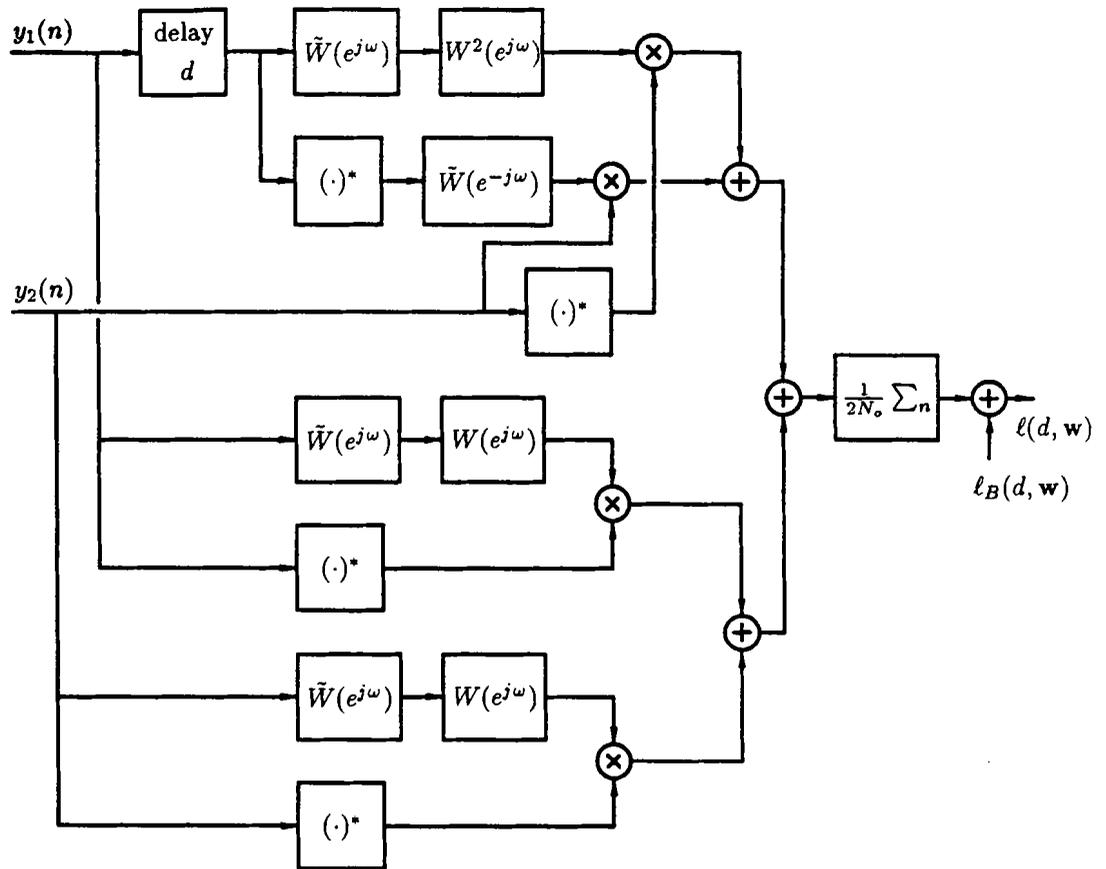


Fig. 2.5 Blockdiagram of an approximate noncausal joint maximum likelihood receiver

2.3.2.3 Adaptive Maximum Likelihood Estimation

The open-loop estimator described above can, in theory, be made adaptive in several ways. This is desirable because the number of parallel receivers, in the open-loop estimator, would clearly be too large for any practical channel $h(n)$. Iterative search procedures, based on different forms of descent algorithm, can be used for the computation of local solutions [31]. These algorithms can also form the basis of suboptimum processors, for on-line estimation of \hat{D} and \hat{h} .

Consider the noncausal joint ML receiver of Fig. 2.5. This receiver computes the likelihood function for a block of data, which is assumed large compared to the time necessary for the system transients to die out. It makes use of noncausal filters, i.e. at any iteration n , the output of the receiver is function of future input data. The estimation can be of the block type, in which the likelihood function is computed for fixed blocks of data and

the estimated values updated on a block-by-block basis. Within the i^{th} block, the values of D and \mathbf{h} can be estimated by performing an exhaustive search independent of the values estimated in the previous blocks, or by performing a limited search, based on some of the information obtained previously. Since the likelihood function for the i^{th} block, denoted $\ell^{(i)}(d, \mathbf{w})$, is generally multimodal with respect to both d and \mathbf{w} , the latter procedure is preferable. Because $\ell^{(i)}(d, \mathbf{w})$ is a random variable, the search should perform a form of average over the blocks. The update formulas can take the form of a general joint algorithm

$$\begin{aligned}\mathbf{w}^{(i+1)} &= f(\mathbf{w}^{(i)}, \ell^{(i)}(d, \mathbf{w})) \\ d^{(i+1)} &= g(d^{(i)}, \ell^{(i)}(d, \mathbf{w})),\end{aligned}\tag{2.25}$$

where the functionals $f(\cdot)$ and $g(\cdot)$ are updating directions. These functionals may be defined, for example, as

$$\begin{aligned}f(\mathbf{w}^{(i)}, \ell^{(i)}(d, \mathbf{w})) &= \max_{\mathbf{w}} E[\ell^{(i)}(d, \mathbf{w})] \quad \text{for } \mathbf{w} \in R_{\mathbf{w}}(i+1) \\ g(d^{(i)}, \ell^{(i)}(d, \mathbf{w})) &= \max_d E[\ell^{(i)}(d, \mathbf{w})] \quad \text{for } d \in R_d(i+1),\end{aligned}\tag{2.26}$$

where the parameter ranges $R_{\mathbf{w}}(i+1)$ and $R_d(i+1)$ are defined in relation with $\mathbf{w}^{(i)}$ and $d^{(i)}$ respectively, in order to narrow down the range of possible values for $(d^{(i+1)}, \mathbf{w}^{(i+1)})$. The information from the previous block is therefore utilized to limit the range of parameter search in the actual block.

Another definition for the functional could be

$$\begin{aligned}f(\mathbf{w}^{(i)}, \ell^{(i)}(d, \mathbf{w})) &= \mathbf{w}^{(i)} + \mu \nabla_{\mathbf{w}} E[\ell^{(i)}(d, \mathbf{w})] \\ g(d^{(i)}, \ell^{(i)}(d, \mathbf{w})) &= d^{(i)} + \alpha \frac{\partial E[\ell^{(i)}(d, \mathbf{w})]}{\partial d},\end{aligned}\tag{2.27}$$

where μ and α are small positive gain factors. This algorithm is a form of block joint steepest-descent algorithm applied on the likelihood function [31]. Note that the derivative information is *added* to the previous estimate value since the objective function $\ell^{(i)}(d, \mathbf{w})$ must be maximized.

The receiver of Fig. 2.5 can be made causal by delaying the two input signals by a suitable number of samples. In this case, the likelihood function at iteration n , denoted $\ell_n(d, \mathbf{w})$, can be computed by using data only available at this time, and a *sample-by-sample* search can be performed. It can be of the form

$$\begin{aligned}\mathbf{w}_{n+1} &= \mathbf{w}_n + \mu \nabla_{\mathbf{w}_n} E[\ell_n(d, \mathbf{w})] \\ d_{n+1} &= d_n + \alpha \frac{\partial E[\ell_n(d, \mathbf{w})]}{\partial d},\end{aligned}\tag{2.28}$$

where a joint steepest-descent search is used to update the estimates at every iteration. This algorithm should converge asymptotically to a solution corresponding to a local maximum

of the objective function. It has also the potential to track the variations of the parameters with time.

Another form of adaptive ML estimator can be based on an hybrid system, in which a coarse open-loop block search is first performed and is followed by a closed-loop search, around the values estimated in the open-loop search [43].

2.3.3 Discussion

Different forms of the joint maximum likelihood estimator, for time-invariant reference delay and filter, have been derived in the above subsections. Every one of these forms is, without exception, difficult to implement. They involve the solution of integral equations, and the number of components in the vector \mathbf{w} complicates even more any joint open-loop estimator. A closed-loop (adaptive) estimator reduces considerably the latter problem, at the expense of introducing convergence inaccuracies (convergence to local solutions). A hybrid system appears to be the best solution, at least conceptually. But the complexity inherent to the receiver of Fig. 2.5, and the computation in real time of the bias term remain problematic.

Nevertheless, the structure of the ML receiver is of interest. First of all, note that if the reference filter is absent, the receiver reduces to a cross-correlation receiver identical to the ML estimator for pure time delay estimation. When the reference filter has to be estimated, the joint ML receiver performs three distinct functions. First, it delays and filters the received signal $y_1(n)$ before it correlates it with $y_2(n)$. Secondly, it performs two extra correlations, in the lower two branches of the receiver. Finally it computes and adds the bias term. Considering only the first function, the form of the receiver is that of a delay element in cascade with a group of filters, both applied on one of the received signals, followed by a comparison (correlation) with the other received signals. This form is appealing and can be retained in other types of joint estimators.

It seems therefore appropriate to consider simpler joint estimators based on different criteria and exhibiting the aforementioned form. These more practical estimators are the subject of the next section, as well as the main subject of this thesis.

2.4 The Joint MMSE and LS Estimators

Taking into considerations the previous discussion, a form for the joint adaptive estimators, based on the MMSE or the LS estimation criteria, is readily obtained. It is

composed of an adaptive branch, with an adaptive delay element connected in cascade with and adaptive filter, and of a reference branch used to generate an error signal. The adaptive branch is either in Type I or in Type II configuration, and is used to estimate jointly the reference delay D_n and the reference filter $h(n)$, or their inverses. If the reference branch is estimated, the configuration is the cancellation one, illustrated in Fig. 2.6. If the inverse of the reference branch is desired, the equalization configuration, as shown in Fig. 2.7, is used. In terms of adaptive delay and filter, Figs. 2.8 gives a detailed form of a Type I joint estimator in cancellation configuration. Note that the cancellation of a certain Type of system (I or II) is always performed by an adaptive system of the same Type, while the equalization is accomplished with the other Type. In the rest of this thesis, whenever it is question of a certain Type of configuration performing a certain task (cancellation or equalization), the system to cancel or equalize (the reference system) is of this Type and it is implicitly assumed that the adaptive system has the proper structure. If it is clear that a specific branch or system (adaptive or reference) is used, then the Type applies to this specific system.

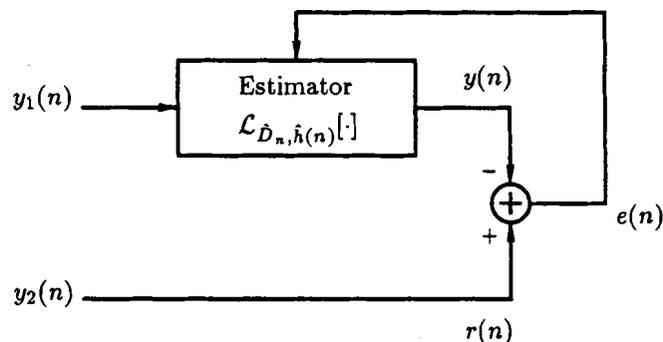


Fig. 2.6 System identification (cancellation) configuration

These joint time delay estimation and adaptive filtering algorithms may be used in any application where both the reference delay and filter must be estimated. They may also find some applications in different areas of adaptive signal processing, especially in the enhancement of already existing techniques involving adaptive filters. The addition of an adaptive delay element to the usual adaptive filtering operations can improve the conventional adaptive parameter estimation techniques that would otherwise be of limited usefulness. In order to appreciate this fact, an adaptive filtering application, in which the input signal and the reference signal exhibit a different sampling rate, is considered in the next subsection.

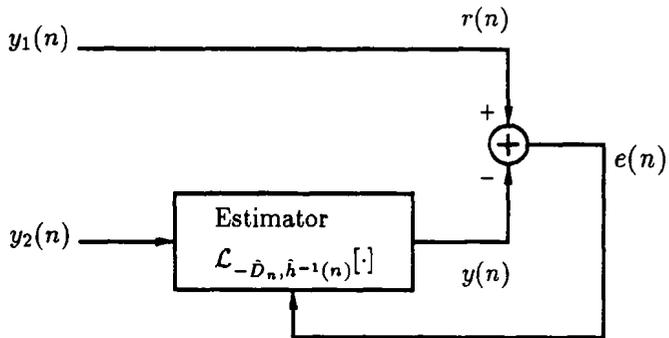


Fig. 2.7 Inverse filtering (equalization) configuration

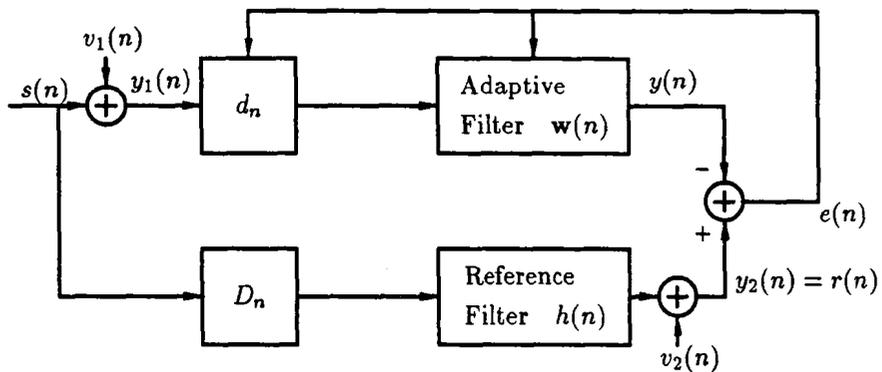


Fig. 2.8 Type I systems in cancellation configuration

2.4.1 The Sampling Rate Difference Problem in Adaptive Filtering

An adaptive system in which the input signal and the reference signal exhibit a different sampling rate may take different forms. One of these possible configurations is given in Fig. 2.9, where noiseless conditions have been assumed. The input signal $s(n)$ and the reference signal $r'(n)$ are sampled at the same rate. A time-variant sampling rate conversion is applied on $r'(n)$, i.e. the uniformly sampled signal $r'(n)$ is ideally interpolated and resampled with a *nonuniform* sampling period $T'(i) = T\lambda(i)$, for $1 \leq i \leq n$ and $\lambda(i)$ a real number.

In such a system, the input signal autocorrelation matrix \mathbf{R} , defined as

$$\mathbf{R} = E[\mathbf{s}(n)\mathbf{s}^H(n)] \quad (2.29)$$

$$\mathbf{s}(n) = [s(n), s(n-1), \dots, s(n-M+1)]^T \quad (2.30)$$

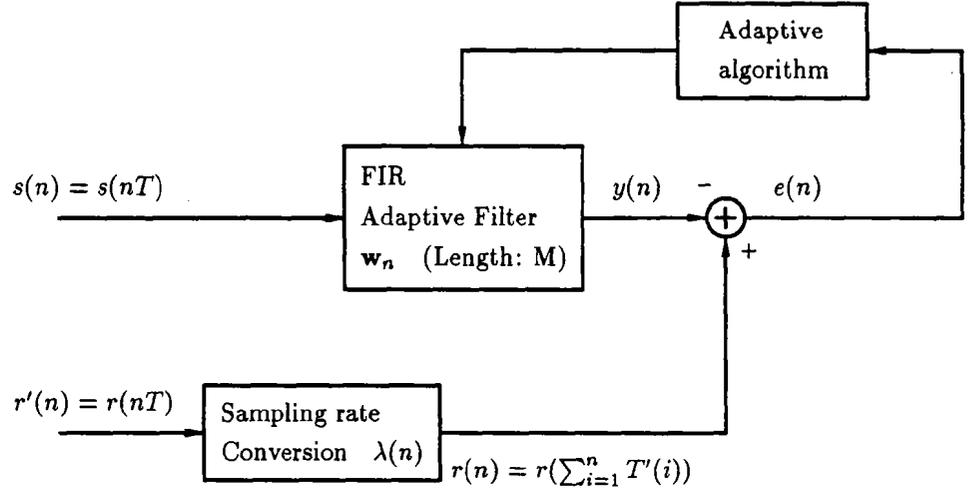


Fig. 2.9 Adaptive system with sampling rate conversion

is constant for a stationary input signal. The cross-correlation vector \mathbf{p}_n , defined as

$$\begin{aligned} \mathbf{p}_n &= E[\mathbf{s}(n)r^*(n)] \\ &= \begin{bmatrix} E[s(n)r^*(n)] \\ E[s(n-1)r^*(n)] \\ \vdots \\ E[s(n-M+1)r^*(n)] \end{bmatrix}, \end{aligned} \quad (2.31)$$

is a function of time. This is the case since the $(l+1)^{th}$ component of \mathbf{p}_n is given by

$$p_{l+1}(n) = \phi_{sr}(n-l, n) = E[s(n-l)r^*(n)] \quad (2.32)$$

$$\begin{aligned} &= E[s(nT-lT)r^*(\sum_{i=1}^n T'(i))] \\ &= \phi_{sr}[(n-l)T - \sum_{i=1}^n T'(i)], \end{aligned} \quad (2.33)$$

where $\phi_{sr}(\tau)$ is the *continuous* complex cross-correlation function between the jointly stationary *continuous* signals $s(t)$ and $r(t)$ and $T'(i)$ is the reference branch sampling period at the i^{th} sampling instant. The continuous cross-correlation function is defined as

$$\phi_{sr}(\tau) = E[s(t)r^*(t-\tau)], \quad (2.34)$$

and, for wide sense stationary signals, is a function of τ only [45]. Equation (2.33) can be written as

$$p_{l+1}(n) = \phi_{sr}[\{n-l - \sum_{i=1}^n \lambda(i)\}T], \quad (2.35)$$

which shows clearly the dependence of \mathbf{p} on n . If $\lambda(i) = 1$ for all i , there is no sampling rate conversion and \mathbf{p} is not time-varying. This shows that even if the sequences $s(n)$ and $r(n)$ are individually stationary (when $\lambda(i)$ is a constant for all i), they are not jointly stationary when the sampling period ratio λ is different from one.

Using the notation of [7], the output of the adaptive filter is defined as

$$y(n) = \mathbf{w}_n^H \mathbf{s}(n). \quad (2.36)$$

The MSE function, defined as $\xi_n = E[|e(n)|^2]$, is then of the form

$$\xi_n = \phi_{rr}(0) + \mathbf{w}_n^H \mathbf{R} \mathbf{w}_n - 2\text{Re}[\mathbf{w}_n^H \mathbf{p}_n], \quad (2.37)$$

where $\phi_{rr}(0)$ is the reference signal variance. Considering Wiener filter theory [43], the weight vector minimizing the MSE at time n is [7]

$$\mathbf{w}_{\text{opt}}(n) = \mathbf{R}^{-1} \mathbf{p}_n. \quad (2.38)$$

The MMSE weight vector is obviously time varying, i.e. the quadratic performance surface is time-variant. Because the matrix \mathbf{R} is constant, its eigenvalues and eigenvectors are constant and the quadratic performance surface is constant in shape, but varies its position with time. If, for example, the sampling rate ratio λ is constant and different from one, and if it is assumed that $\phi_{sr}(\tau) \rightarrow 0$ as $\tau \rightarrow \infty$, then $\lim_{n \rightarrow \infty} \mathbf{p}_n = 0$ and $\mathbf{w}_{\text{opt}}(n) \rightarrow 0$ as $n \rightarrow \infty$. This particular case illustrates the limiting situation where the filter input and the reference signal are totally decorrelated and the adaptive filter is virtually useless. A similar situation happens when the adaptive filter time span is larger than the maximum time lag for which the filter input and the reference signal are correlated.

This decorrelation between $s(n)$ and $r(n)$ is equivalent to a time-varying delay, which can be computed as follows. Assume that for some integers M and K , the following relation is true

$$KT = \sum_{i=1}^M T'(i) = T \sum_{i=1}^M \lambda(i), \quad (2.39)$$

i.e. $r(n)$ and $s(n)$ are time-aligned at time KT . Then, for $n = K + l$, $r(n)$ is

$$r(K + l) = r(KT + T \sum_{i=1}^l \lambda(K + i)) \quad (2.40)$$

and $s(n)$ is

$$s(K + l) = s(KT + lT). \quad (2.41)$$

Then $r(n)$ lags $s(n)$ by the time-varying value $D_n = T(l - \sum_{i=1}^l \lambda(K+i))$. An additional adaptive delay element, connected in Type I or in Type II configuration with the adaptive filter, can therefore make viable the original adaptive solution by compensating for the sampling rate difference. An adaptive system, in cancellation mode and corresponding to a Type I configuration, is illustrated in Fig. 2.10. Note that the reference branch in Fig. 2.9 is of Type II. Note also that if the reference delay D_n is constant with time (i.e. the sampling rates are the same), the two types of systems are equivalent [†].

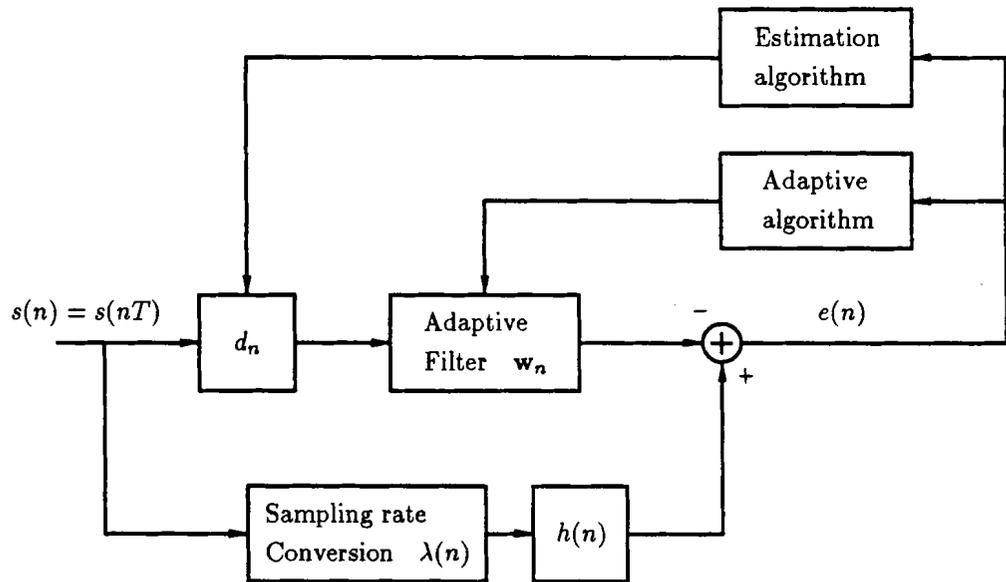


Fig. 2.10 Type I adaptive system with sampling rate conversion

2.4.2 Discussion

The form of joint MMSE or LS estimators that is favoured in this thesis has been introduced. It has the advantage to be very simple since it essentially mimics the form of the operator $\mathcal{L}_{D_n, h(n)}(\cdot)$. Its basis is the conventional adaptive filter, using the MMSE or the LS estimation criteria. The combination of an adaptive delay element and an adaptive filter constitutes by itself a joint delay estimation and channel identification technique that can be compared to any other form of such joint estimator, in particular the joint ML estimator

[†] If the sampling rates are different, the sampling rate conversion is equivalent to a linear time-variant system and such systems are not, in general, commutative. Types I and II are therefore not equivalent in this general case.

derived in Section 2.3. But it constitutes also an improved version of the conventional adaptive filter, which increases its potential utility.

2.5 Summary

Three structures for performing joint time delay estimation and channel identification have been presented. A mathematical model for the received pair of signals has been introduced. The joint maximum likelihood estimator for Gaussian signals has been derived and its limited practical utility discussed. The ML estimator has been used to specify a simpler joint estimator structure, composed of an adaptive delay element operated in conjunction with an adaptive filter. The MMSE and the LS estimation criteria are well suited for that new structure. It was finally noted that the joint delay estimation and adaptive filter algorithm can also be considered as an enhanced version of the conventional adaptive filter.

Joint Time Delay Estimation and Adaptive Chapter 3 Minimum Mean Squared Error Filtering: The Joint Steepest-Descent Algorithm

3.1 Introduction

This chapter presents an analysis of joint delay estimation and channel identification based on the minimum mean squared error (MMSE) performance index, when the channel identification is specifically performed by an adaptive transversal filter and the delay estimation is accomplished independently from this filter, by an adaptive delay element. A joint steepest-descent algorithm is investigated here and a joint LMS algorithm will be considered in Chapter 4.

The principal contributions of these two chapters are the generalization of existing gradient-based time delay estimation without the reference filter $h(n)$, and the analysis of a new joint algorithm for the synchronization of the input and the reference signals used by an adaptive filter. The joint steepest-descent and LMS algorithms are generalizations of joint clock phase recovery and adaptive equalization based on MMSE phase tracking. This generalization is based on the facts that the sampling period *and* the sampling phase are tracked, and that the signals considered are general and not limited to data signals. These joint algorithms assume generally that the input signal and the reference signal fed to an adaptive filter are not sampled with the same clock period. They also allow the tracking of time-varying delays, in the reference path, by a process separated from the adaptive filter, which itself is free to perform the task of modeling the linear filter $h(n)$ or its inverse. The material presented here and in Chapter 4 expands upon the work published in [13] and [29].

The chapter is structured as follows. Some general theoretical concepts are presented in Section 3.2. In particular, the minimum mean squared (MSE) function is derived in general terms and a derivative-based search of its minimum, with respect to the adaptive delay, is discussed. These general concepts are then applied in Section 3.3, where the joint SD algorithm is considered in some details. Finally, the theoretical results derived in Sections 3.2 and 3.3 are applied to some special cases in Section 3.4.

3.2 General MMSE Theory

Recall that the model studied is (see Section 2.2)

$$\begin{aligned} y_1(n) &= s(n) + v_1(n) \\ y_2(n) &= \mathcal{L}_{D_n, h(n)}[s(n)] + v_2(n). \end{aligned} \tag{3.1}$$

Recall also that, depending on the problem at hand, the operator $\mathcal{L}_{D_n, h(n)}[s(n)]$ can take the form of the filtering of a delayed version of $s(n)$ or the form of a filter followed by a delay. The former configuration is defined as a Type I system and the latter as a Type II system. These two definitions also apply to the joint adaptive estimator. Note that the two types of systems are equivalent if the corresponding delay is constant with time.

The adaptive filter is a transversal filter, with a weight vector \mathbf{w}_n of length M . The goal of this filter is to estimate the impulse response $h(n)$ or its inverse. It is desired that the reference delay value D_n be estimated separately from the adaptive filter, by an adaptive delay element d_n cascaded with the filter in Type I or Type II form. In joint MMSE delay estimation and adaptive filtering, the mean squared error surface is searched by both the adaptive filter estimation algorithm and the delay estimation algorithm. In system identification (cancellation) scenarios, $y_1(n)$ is filtered by an estimate of $\mathcal{L}_{D_n, h(n)}[s(n)]$ and the resulting signal is subtracted from $y_2(n)$, in order to form the error signal. In inverse filtering (equalization), $y_2(n)$ is passed through an estimate of $\mathcal{L}_{-D_n, h^{-1}(n)}[s(n)]$ and compared to $y_1(n)$. This was illustrated in Figs. 2.6 and 2.7.

3.2.1 The Mean Squared Error Function

In general, the output of the adaptive branch can be defined as $y(n)$ and the reference signal as $r(n)$. Then the error signal is defined as

$$e(n) = r(n) - y(n), \tag{3.2}$$

and the MSE function, at time n , as

$$\xi_n = E[|e(n)|^2]. \quad (3.3)$$

The joint estimation can be thought of as taking place in a vector space made of a weight vector subspace and a delay subspace. The two subspaces are not orthogonal, which implies that the two estimation processes are not independent (because the adaptive filter can model a reference delay).

The MSE function, for all possible combinations of configurations (cancellation or equalization in Type I or Type II), can be represented by a general expression. In order to do so, define as $u(n)$ the input to the adaptive *branch*, whether this branch is in Type I or Type II configuration. The output of the adaptive branch is $y(n)$ and the reference signal is $r(n)$. This is illustrated in Fig. 3.1.

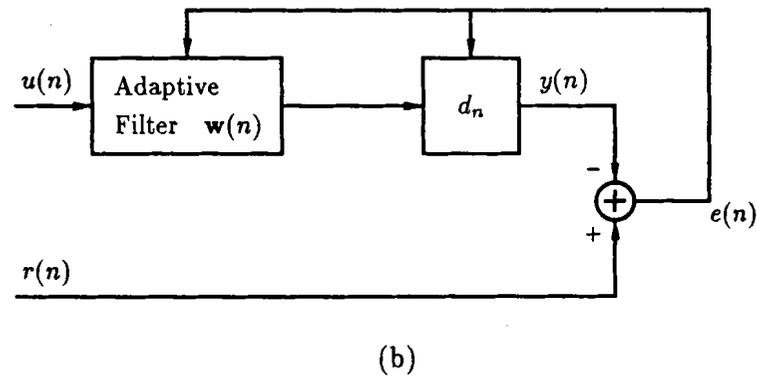
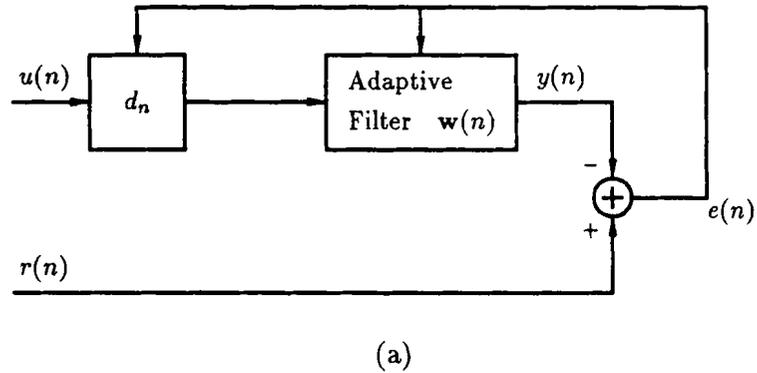


Fig. 3.1 General model for (a) a Type I adaptive system and for (b) a Type II adaptive system

Assume also that a correlation function $\phi_{ab}(n, m)$, between two discrete signals $a(n)$ and $b(m)$, is defined in terms of the correlation function between the continuous signals $a(t)$ and $b(t)$ as

$$\begin{aligned}\phi_{ab}(n, m) &= E[a(n)b^*(m)] \\ &= E[a(nT + \Delta_n)b^*(mT + \Delta_m)] \\ &= \phi_{ab}(nT + \Delta_n, mT + \Delta_m),\end{aligned}\tag{3.4}$$

where Δ_n and Δ_m are the delays imposed on the continuous signals at iterations n and m †.

The MSE function can then be represented by either one of the following equivalent equations

$$\xi_n = \begin{cases} \phi_{rr}(n, n) + \mathbf{w}_{dn}^H \mathbf{R}_n \mathbf{w}_{dn} - 2\text{Re}[\mathbf{w}_{dn}^H \mathbf{p}_n] \\ \phi_{rr}(n, n) + \phi_{yy}(n, n) - 2\text{Re}[\phi_{yr}(n, n)], \end{cases}\tag{3.5}$$

where $\text{Re}[\cdot]$ is the real value operator, $\phi_{rr}(n, m)$ and $\phi_{yy}(n, m)$ are respectively the autocorrelation functions of the reference signal and the adaptive branch output, $\phi_{yr}(n, m)$ is the cross-correlation function between this output and the reference signal, \mathbf{R}_n is the autocorrelation matrix of a delayed version of the adaptive branch input $u(n)$ and \mathbf{p}_n is the cross-correlation vector between the same delayed input and the reference signal. Finally, \mathbf{w}_{dn} is a delayed version of the weight vector \mathbf{w}_n .

The autocorrelation matrix and the cross-correlation vector are then expressed as

$$\mathbf{R}_n = E[\mathbf{u}_n \mathbf{u}_n^H]\tag{3.6}$$

$$\mathbf{p}_n = E[\mathbf{u}_n r^*(n)],\tag{3.7}$$

where \mathbf{u}_n is the equivalent vector of delayed input samples, stored at iteration n , in the adaptive filter delay line. For a Type I adaptive system, this vector is

$$\mathbf{u}_n = [u(nT - d_n), u(nT - T - d_{n-1}), \dots, u(nT - MT + T - d_{n-M+1})]^T.\tag{3.8}$$

For a Type II system, \mathbf{u}_n is

$$\mathbf{u}_n = [u(nT - d_n), u(nT - T - d_n), \dots, u(nT - MT + T - d_n)]^T.\tag{3.9}$$

Similarly, the weight vector is given by

$$\mathbf{w}_{dn} = \begin{cases} \mathbf{w}_n = \mathbf{w}(nT) & \text{Type I} \\ \mathbf{w}_{nT-d_n} = \mathbf{w}(nT - d_n) & \text{Type II.} \end{cases}\tag{3.10}$$

† The difference between a discrete and a continuous correlation function is not explicitly denoted otherwise than by using discrete or continuous time arguments.

Note that, as with all signals and systems, the adaptive filter transfer function is assumed strictly bandlimited to $-\pi < \omega < \pi$. The vector $\mathbf{w}(nT - d_n)$ is therefore obtained by resampling at $nT - d_n$ the continuous version of $\mathbf{w}(nT)$. Note also that the above relationships are true if the output of the adaptive branch is defined as

$$y(n) = \mathbf{w}_{dn}^H \mathbf{u}_n. \quad (3.11)$$

Some other variations of Type I and II *adaptive* systems, for which the MSE function form of (3.5) applies, can also be defined. For example, a modification of a Type I system is one in which the delay d_n propagates instantaneously through the adaptive filter delay line, i.e. where \mathbf{u}_n is represented as in (3.9). In Type II configuration, it is possible to transfer the adaptive delay to the reference branch. For the cancellation of a Type II configuration, this means that a *negative delay* d_n is applied in the reference branch, instead of a *positive delay* d_n in the adaptive filter branch. Such a system is illustrated in Fig. 3.2 and is called a Type II-DRB (delay in reference branch) system. For the equalization of a Type I configuration, the adaptive delay can be made positive in the reference branch, instead of being negative in the adaptive branch. These particular adaptive Type II-DRB configurations have the advantage that $\mathbf{w}_{dn} = \mathbf{w}_n$ and will be preferred in practice. The Type II adaptive system with a delay in the adaptive branch is called a Type II-DAB adaptive system. Note that a signal $s(n)$ that enters a delay d_n always becomes $s(nT - d_n)$, and it is the sign of d_n that indicates if the signal is retarded (positive sign) or advanced (negative sign). Finally, note that a negative delay is always implemented as a portion of a *positive* reference delay, and corresponds to a decrease of this reference delay.

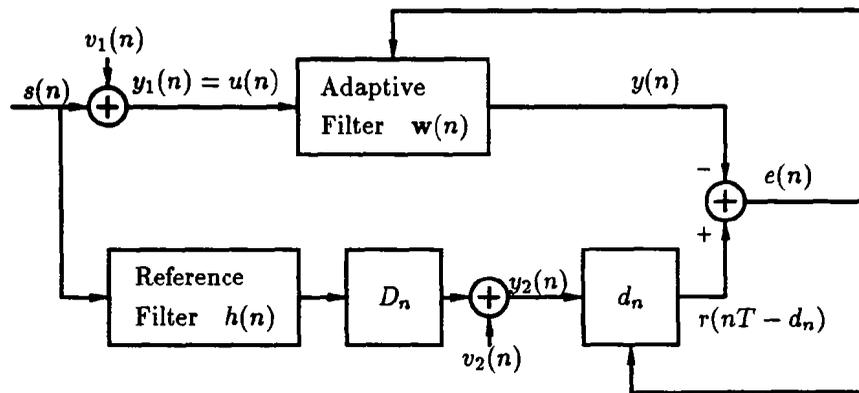


Fig. 3.2 Type II systems, with negative delay, in cancellation configuration

The two forms of ξ_n , given in (3.5), reflect the nature of the joint estimator operation. In the weight vector subspace, associated to the first equation of (3.5), the MSE function is a quadratic surface [7]. The one-dimensional delay subspace is naturally linked to the correlation functions of the second equation of (3.5). The MSE function is not, in general, unimodal with respect to d_n . In order to see this, note that ξ_n depends on correlation functions that vary according to the adaptive filter and the operator $\mathcal{L}[s(n)]$, as well as to the autocorrelation function of the signals $u(n)$ and $r(n)$. All of these functions are multimodal with respect to their time argument, which in turn causes the MSE function to behave similarly with respect to d_n and produces a multitude of local extrema.

3.2.1.1 The MSE Function for Specific Configurations

The MSE function is explicitly derived below, for the two Types of joint adaptive configurations. The resulting expressions are instructive in that they show the relationship between the adaptive filter coefficients and the different correlation functions involving the time-varying delays. Note that the derivations are performed as functions of the general signals $u(n)$, $y(n)$ and $r(n)$ defined above, and apply to both the system identification (cancellation) and inverse filtering (equalization) configurations.

Type I Adaptive Configuration

Using the second equation of (3.5), the MSE function is

$$\begin{aligned} \xi_n = & \phi_{rr}(n, n) + \sum_i \sum_j w_{ni}^* w_{nj} \phi_{uu}(nT - iT - d_{n-i}, nT - jT - d_{n-j}) \\ & - 2\text{Re}[\sum_i w_{ni}^* \phi_{ur}(nT - iT - d_{n-i}, nT)], \end{aligned} \quad (3.12)$$

where w_{ni} is the i^{th} component of the adaptive filter weight vector w_n at time n .

Type II-DAB Adaptive Configuration

The MSE function is

$$\begin{aligned} \xi_n = & \phi_{rr}(n, n) + \sum_i \sum_j w_{(nT-d_n)i}^* w_{(nT-d_n)j} \phi_{uu}(nT - iT - d_n, nT - jT - d_n) \\ & - 2\text{Re}[\sum_i w_{(nT-d_n)i}^* \phi_{ur}(nT - iT - d_n, nT)], \end{aligned} \quad (3.13)$$

where $w_{(nT-d_n)i}$ is the i^{th} component of the delayed adaptive filter.

Note the effect of the adaptive delay in these two configurations, in particular in the Type II structure, where the adaptive filter coefficients are directly affected by the delay.

Type II-DRB Adaptive Configuration

In a modified Type II structure, as shown in Fig. 3.2, the delay is applied on the reference signal only and the MSE function is of the form

$$\begin{aligned} \xi_n = & \phi_{rr}(nT - d_n, nT - d_n) + \sum_i \sum_j w_{ni}^* w_{nj} \phi_{uu}(nT - iT, nT - jT) \\ & - 2\text{Re}[\sum_i w_{ni}^* \phi_{ur}(nT - iT, nT - d_n)]. \end{aligned} \quad (3.14)$$

The above expressions will be applied, in the subsequent sections, to the mathematical model of (3.1), used in the cancellation and equalization configurations.

3.2.2 Derivative-Based Delay Estimation

As argued in Subsection 3.2.1, the MSE function is multimodal with respect to the delay d_n (consider (3.12) to (3.14)). This causes a problem in the search for the minimum of ξ_n with respect to d_n . In closed-loop estimation, this phenomenon leads to false lock problems, as in phase-locked loops. These problems are generally solved by designing an acquisition procedure, in which the delay estimate is varied until the algorithm falls in its tracking region, near the MSE global minimum. Once in tracking mode, the estimation algorithm can compute the derivative of the MSE function with respect to the delay value, and generate a correcting signal that brings the loop into lock. This is the essence of most closed-loop MMSE methods proposed for the simple signal model in which

$$\mathcal{L}_{D_n, h(n)}[s(n)] = s(nT - D_n). \quad (3.15)$$

A general form of the derivative-based delay estimation algorithm can be such that d_n is updated using a function $f(\cdot)$ of the previous delay estimate values, as well as a function of the MSE surface. This form can be expressed as

$$d_{n+1} = f(d_n) - g[\gamma_n(d_n)], \quad (3.16)$$

where $\gamma_n(\cdot)$ represents the MSE function or an estimate of it at time n and $g(\cdot)$ is a functional that effectively computes a form of derivative of $\gamma_n(\cdot)$, with respect to d_n . Note that $\gamma_n(\cdot)$ is a function of n not only through d_n , but also through w_n and $h(n)$. The form of (3.16) is motivated by existing recursive optimization algorithms [31] or recursive system identification algorithms [4]. Assume that $f(\cdot)$ and $g(\cdot)$ are real coefficients difference

equations of the form

$$f(d_n) = \sum_{j=0}^J c_j d_{n-j} \quad (3.17)$$

$$g[\rho_n(d_n)] = \sum_{i=0}^I \alpha_i \frac{\partial \rho_{n-i}}{\partial d_{n-i}}. \quad (3.18)$$

Then, d_n is updated using a filtered version of the previous estimate values, as well as a filtered version of the previous derivatives. This form is

$$d_{n+1} = \sum_{j=0}^J c_j d_{n-j} - \sum_{i=0}^I \alpha_i \frac{\partial \gamma_{n-i}(d_{n-i})}{\partial d_{n-i}}. \quad (3.19)$$

In a first-order algorithm, $c_0 = 1$, $\alpha_0 = \alpha$ and all the other coefficients are zero. This transforms (3.19) into the steepest-descent or the LMS algorithm, having the form

$$d_{n+1} = d_n - \alpha \frac{\partial \gamma_n(d_n)}{\partial d_n}. \quad (3.20)$$

A common assumption in the analysis of tracking algorithms is that the estimate is close to the optimum value, which allows the linearization of the tracking loop [46], [29]. The Taylor expansion of $\gamma_n(d_n)$, around $d_n = \Theta_n$, is

$$\begin{aligned} \gamma_n(d_n) &= \sum_{i=0}^{\infty} \frac{\partial^{(i)} \gamma_n(d_n)}{\partial d_n^i} \Big|_{d_n=\Theta_n} \frac{(d_n - \Theta_n)^i}{i!} \\ &= \gamma_n(\Theta_n) + (d_n - \Theta_n) \dot{\gamma}_n(\Theta_n) + 1/2 (d_n - \Theta_n)^2 \ddot{\gamma}_n(\Theta_n) + \dots, \end{aligned} \quad (3.21)$$

where the dot denotes the derivative with respect to d_n . Assuming that Θ_n is close to a minimum (local or global) of the MSE function estimated by $\gamma_n(d_n)$, the higher terms are neglected and the error function can be expressed as

$$\gamma_n(d_n) \approx \gamma_n(\Theta_n) + (d_n - \Theta_n) \dot{\gamma}_n(\Theta_n) + 1/2 (d_n - \Theta_n)^2 \ddot{\gamma}_n(\Theta_n). \quad (3.22)$$

This approximation is used in order to linearize the delay estimation algorithm. In delay tracking conditions, the linearized general algorithm is obtained by combining (3.19) and (3.22), and assuming that $\dot{\gamma}_n(\Theta_n) \approx 0$ [†]. It is given by

$$d_{n+1} = \sum_{j=0}^J c_j d_{n-j} + \sum_{i=0}^I \alpha_i \ddot{\gamma}_{n-i}(\Theta_{n-i}) (\Theta_{n-i} - d_{n-i}). \quad (3.23)$$

[†] Note that this assumption is true when $\gamma_n(d_n)$ is the MSE function and Θ_n is a minimum, but that it can be false if a stochastic approximation of the MSE function is used.

The linearized first-order algorithm is

$$d_{n+1} = d_n - \alpha \ddot{\gamma}_n(\Theta_n)(d_n - \Theta_n), \quad (3.24)$$

which can be written as

$$d_{n+1} = (1 - \alpha \ddot{\gamma}_n(\Theta_n))d_n + \alpha \Theta_n \ddot{\gamma}_n(\Theta_n). \quad (3.25)$$

Equation (3.25) models the behaviour of a first-order delay-lock loop [46]. The variations of Θ_n represent the variations of the minimum tracked by the loop and $\ddot{\gamma}_n(\Theta_n)$ represents the *loop delay error generator characteristic* for that minimum [19], i.e. the function of Θ_n by which the loop error is multiplied. At iteration n , (3.25) approaches the closest minimum Θ_n if $|1 - \alpha \ddot{\gamma}_n(\Theta_n)| < 1$, i.e. if

$$0 < \alpha < \frac{2}{\ddot{\gamma}_n(\Theta_n)}. \quad (3.26)$$

3.2.2.1 A Restricted Class of First-Order Delay Tracking Algorithm

The expression (3.25) is a linear difference equation with time-varying coefficients, which makes difficult any convergence and stability studies. It is a function of the variations with time, of both the error function $\gamma_n(\cdot)$ and the value Θ_n . A restricted class of problem allows the derivation of useful results. In this class, it is assumed that the function $\ddot{\gamma}_n(\Theta_n)$ is constant. This assumption implies that the delay error generator characteristic is not influenced by the adaptive or reference filters changing characteristics, nor it is by Θ_n . Then, the first-order difference equation has for solution

$$d_n = (1 - \alpha \ddot{\gamma})^n d_0 + \alpha \ddot{\gamma} \sum_{\ell=0}^{n-1} (1 - \alpha \ddot{\gamma})^\ell \Theta_{n-\ell-1}. \quad (3.27)$$

Equation (3.27) converges if $|1 - \alpha \ddot{\gamma}| < 1$, i.e. for

$$0 < \alpha < 2/\ddot{\gamma}. \quad (3.28)$$

The time constant of delay adaptation can be defined by fitting the geometric ratio $1 - \alpha \ddot{\gamma}$ to an exponential with time constant τ_{del}

$$\begin{aligned} 1 - \alpha \ddot{\gamma} &= e^{-1/\tau_{\text{del}}} \\ &\approx 1 - 1/\tau_{\text{del}}. \end{aligned}$$

The time constant of delay adaptation is therefore

$$\tau_{\text{del}} \approx \frac{1}{\alpha \ddot{\gamma}}. \quad (3.29)$$

3.2.3 Discussion

Some general results have been established in this section. The effect of the adaptive delay on the MSE function was shown to be dependent on the structure used in the estimation. If a Type I *adaptive* system (delay in front of filter) is such that the delay d_n propagates instantaneously through the adaptive delay line, or if the adaptive delay is transferred to the reference branch in a Type II-DRB *adaptive* configuration (delay after filter), then simplified expressions result. By using a truncated Taylor expansion of an estimate of the MSE function, it is possible to obtain general results about the adaptive delay steepest-descent algorithm. In particular, by restricting the second derivative of the MSE function estimate to be constant, the gain factor range insuring convergence of the adaptive delay SD algorithm can be computed, as well as the algorithm time constant. This special case is not too restrictive and is applicable to systems in which the reference filter $h(n)$ varies slowly, in tracking mode. These results will be used in the subsequent sections with the function $\gamma_n(\cdot)$ specific to the joint steepest-descent algorithm.

3.3 The Joint Steepest-Descent Algorithm

The simplest joint derivative-based algorithm is the first order one, which is composed of the usual steepest-descent (SD) adaptive filter, of the form

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla_{\mathbf{w}_{dn}} \xi_n, \quad (3.30)$$

and of the SD adaptive delay algorithm, expressed as

$$d_{n+1} = d_n - \alpha \frac{\partial \xi_n}{\partial d_n}. \quad (3.31)$$

Note that (3.31) is just equation (3.20) with

$$\gamma_n(d_n) = \xi_n. \quad (3.32)$$

The combination of (3.30) and (3.31) allows some extra flexibility in the application of the joint SD algorithm. Define $\xi\{d_n, \mathbf{w}_{dn}\}$ as the MSE function for specific values of the adaptive delay and weight vector. Then the adjustments of d_n and \mathbf{w}_n can be based both on $\xi\{d_n, \mathbf{w}_{dn}\}$, giving the following form of joint SD algorithm

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu \nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\} \\ d_{n+1} &= d_n - \alpha \frac{\partial \xi\{d_n, \mathbf{w}_{dn}\}}{\partial d_n}. \end{aligned} \quad (3.33)$$

The adaptive weight vector can be adjusted before the delay adaptation, producing the algorithm

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu \nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\} \\ d_{n+1} &= d_n - \alpha \frac{\partial \xi\{d_n, \mathbf{w}_{d(n+1)}\}}{\partial d_n}, \end{aligned} \quad (3.34)$$

or the delay element can be adjusted before the filter adaptation, giving

$$\begin{aligned} d_{n+1} &= d_n - \alpha \frac{\partial \xi\{d_n, \mathbf{w}_{dn}\}}{\partial d_n} \\ \mathbf{w}_{n+1} &= \mathbf{w}_n - \mu \nabla_{\mathbf{w}_{dn}} \xi\{d_{n+1}, \mathbf{w}_{dn}\}. \end{aligned} \quad (3.35)$$

The algorithms of (3.33) to (3.35) can be generalized even further by allowing repeated adaptations on the same input data, which is referred to as *data reuse* [47]. This offers a large number of possibilities for the alternation of the two adaptive processes. The algorithms (3.33) to (3.35) will be the only ones considered in this chapter and the algorithm (3.33) will be referred to most of the time, when the expression “joint SD algorithm” is used. The two special forms of (3.34) and (3.35) will be called the *joint alternate algorithms*.

The convergence of the joint SD algorithm is considered in the next subsection. Then, Subsection 3.3.2 treats of the delay tracking properties of the algorithm.

3.3.1 Convergence of the Joint SD Algorithm

A necessary condition for a specific d_n and \mathbf{w}_n to be a stationary solution of the algorithms (3.33) to (3.35) is that both of the following equations be satisfied [37]

$$\begin{cases} \nabla_{\mathbf{w}_{dn}} \xi_n = 0 \\ \frac{\partial \xi_n}{\partial d_n} = 0. \end{cases} \quad (3.36)$$

This condition is general and applies to any type of adaptive structure. Note that the first equation of (3.36) is in fact a *necessary and sufficient* condition for convergence. This is so because ξ_n is quadratic with respect to \mathbf{w}_{dn} , which implies that there is a unique minimum in \mathbf{w}_{dn} , for a given value d_n . When the first equation of (3.36) is satisfied, this unique solution is attained, and any further modifications of d_n will increase ξ_n . This is the case because the adaptive filter models both the relative delay and the reference filter in the minimum MSE sense. Then, this solution corresponds also to a minimum with respect to d_n . The sufficiency of the condition is due to the uniqueness of the minimum with respect to \mathbf{w}_{dn} . A better idea of the convergence properties of the joint SD algorithm can be obtained by assuming a particular Type I or Type II structure.

3.3.1.1 Convergence Results for Particular Structures

Assume an adaptive Type I configuration in which the delay d_n propagates instantaneously through the adaptive filter delay line, or an adaptive Type II-DRB configuration in which the adaptive delay is applied to the reference branch, as in Fig. 3.2. In this case, the first equation of (3.5) is such that the input signal autocorrelation matrix is constant and the adaptive weight vector is not function of the delay d_n . Furthermore, assuming that the reference filter is stationary, $\phi_{rr}(nT - d_n, nT - d_n) = \phi_{rr}(0)$ and is not function of d_n . The necessary condition of (3.36) reduces to

$$\begin{cases} \mathbf{w}_n = \mathbf{R}^{-1} \mathbf{p}_n \\ \text{Re}[\mathbf{w}_n^H \dot{\mathbf{p}}_n] = 0, \end{cases} \quad (3.37)$$

i.e. the weight vector solution is the Wiener solution when the delay d_n is such that \mathbf{w}_n is *orthogonal* to $\dot{\mathbf{p}}_n$ or the product $\mathbf{w}_n^H \dot{\mathbf{p}}_n$ is purely imaginary. Note that the solution of (3.37) is *not unique*, which constitutes one of the most important characteristic of the joint SD algorithm. This shows again the need for an acquisition algorithm that brings the estimates close to their global optimum, before any tracking algorithm takes over.

The cross-correlation vector \mathbf{p}_n is a function of the cross-correlation function between the delayed input signal and the reference signal. Its components are in fact the samples of the corresponding continuous cross-correlation function. This forces the vector \mathbf{p}_n to follow a path, in the weight vector subspace, specified by the cross-correlation function and makes the weight solution $\mathbf{R}^{-1} \mathbf{p}_n$ a member of a specific subset of the weight vector subspace. In order to see the nature of the solution in the delay subspace, express the MSE function as

$$\xi_n = \phi_{rr}(0) - \mathbf{p}_n^H \mathbf{R}^{-1} \mathbf{p}_n + (\mathbf{w}_n - \mathbf{R}^{-1} \mathbf{p}_n)^H \mathbf{R} (\mathbf{w}_n - \mathbf{R}^{-1} \mathbf{p}_n), \quad (3.38)$$

where the reference filter has been assumed stationary.

If the first condition of (3.37) is respected, the MSE function becomes

$$\begin{aligned} \xi_o(d_n) &= \xi_n |_{\mathbf{w}_n = \mathbf{R}^{-1} \mathbf{p}_n} \\ &= \phi_{rr}(0) - \mathbf{p}_n^H \mathbf{R}^{-1} \mathbf{p}_n. \end{aligned} \quad (3.39)$$

The second condition of (3.37) is respected if d_n is a minimum of $\xi_o(d_n)$, which is function of the cross-correlation between the delayed input signal and reference signal.

Therefore, in order to be a stationary solution, the couple (d_n, \mathbf{w}_n) must be such that d_n is a minimum of $\xi_o(d_n)$ and \mathbf{w}_n is given by $\mathbf{R}^{-1} \mathbf{p}_n$. The convergence toward this solution can be interpreted by considering a small-signal representation of \mathbf{p}_n . First, note that

$$\nabla_{\mathbf{w}_n} \xi_n = 2\mathbf{R} \mathbf{w}_n - 2\mathbf{p}_n. \quad (3.40)$$

Using this expression, the joint gradient algorithm of (3.33) can be expressed as

$$\begin{aligned}\mathbf{w}_{n+1} &= (\mathbf{I} - 2\mu\mathbf{R})\mathbf{w}_n + 2\mu\mathbf{p}_n \\ d_{n+1} &= d_n + 2\alpha\text{Re}[\mathbf{w}_n^H \dot{\mathbf{p}}_n].\end{aligned}\quad (3.41)$$

If the gain constant α is small, the change from d_n to d_{n+1} is likely to be small also and then $\phi_{ur}(-jT \pm d_{n+1})$ (the $(j+1)^{\text{th}}$ component of \mathbf{p}_n) can be approximated as

$$\phi_{ur}(-jT \pm d_{n+1}) \approx \phi_{ur}(-jT \pm d_n) \pm (d_{n+1} - d_n) \left. \frac{\partial \phi_{ur}(\tau)}{\partial \tau} \right|_{\tau=-jT \pm d_n} \quad (3.42)$$

for $0 \leq j \leq M-1$. Note that the plus sign applies to the Type II-DRB case and the minus sign to the special Type I assumed here (the delay propagates instantaneously through the adaptive filter delay line). Then, \mathbf{p}_{n+1} can be approximated as

$$\mathbf{p}_{n+1} \approx \mathbf{p}_n + (d_{n+1} - d_n) \dot{\mathbf{p}}_n. \quad (3.43)$$

Using the second equation of (3.41), equation (3.43) becomes

$$\mathbf{p}_{n+1} \approx \mathbf{p}_n + 2\alpha\text{Re}[\mathbf{w}_n^H \dot{\mathbf{p}}_n] \dot{\mathbf{p}}_n \quad (3.44)$$

and the joint algorithm is then approximately

$$\begin{aligned}\mathbf{w}_{n+1} &= (\mathbf{I} - 2\mu\mathbf{R})\mathbf{w}_n + 2\mu\mathbf{p}_n \\ \mathbf{p}_{n+1} &\approx \mathbf{p}_n + 2\alpha\text{Re}[\mathbf{w}_n^H \dot{\mathbf{p}}_n] \dot{\mathbf{p}}_n.\end{aligned}\quad (3.45)$$

The interpretation is that, as d_n is modified, \mathbf{p}_n moves along a predetermined path (determined by $\phi_{ur}(d_n)$), changing the location of the performance surface minimum, trying to reach the point where the adaptive filter does not need to compensate for any delay. This point is attained when \mathbf{w}_n equals $\mathbf{R}^{-1}\mathbf{p}_n$.

These results give a qualitative description of the convergence behaviour of the joint SD algorithm, independently of the way the two adaptation processes are alternated (i.e. they apply to algorithm (3.33), as well as to the algorithms (3.34) and (3.35) with minor modifications, as long as the special Type I or Type II-DRB structures assumed at the beginning apply). More rigorous results, that apply to the alternate joint algorithms of (3.34) and (3.35), are given next.

3.3.1.2 A Condition of Convergence for the Joint Alternate Algorithm

If the adaptation factors μ and α are chosen sufficiently small, the process always reaches a limit point [48]. In what follows, a condition on μ and α is given, that ensures convergence of the joint alternate algorithms of (3.34) and (3.35), for both Types of systems.

This condition is derived in [37] for joint carrier phase acquisition and adaptive equalization, in digital communications. It is reformulated here for the problem at hand. This condition is general in that it establishes the stability range for the two adaptation factors such that the MSE is reduced at each iteration, for both Type I and Type II systems, no matter how the two adaptive processes are alternated (i.e. data reuse can happen). It is also important because it confirms that, with the right parameters, the joint SD algorithm converges eventually to a stationary point. The condition does not strictly apply to the joint SD algorithm of (3.33), but it gives useful indications about the convergence of this algorithm as well.

First, assume that the adaptation factors can be time-variant and denote them as μ_n and α_n . Express the MSE as an explicit function of d_n and \mathbf{w}_{dn} , i.e. as $\xi\{d_n, \mathbf{w}_{dn}\}$. Define a *stationary point* of $\xi\{d_n, \mathbf{w}_{dn}\}$ as a solution (d_n, \mathbf{w}_{dn}) of the necessary condition (3.36). It is said that $\xi\{d_n, \mathbf{w}_{dn}\}$ converges to a stationary point if, for every $\epsilon > 0$, there is an N such that

$$\left[\frac{\partial}{\partial d_n} \xi\{d_n, \mathbf{w}_{dn}\} \right]^2 + [\nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\}]^H [\nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\}] < \epsilon \quad (3.46)$$

for all $n > N$.

The decrease in MSE due to the n^{th} adjustment is denoted by $\Delta\xi_n$ and is defined as

$$\Delta\xi_n = \xi\{d_n, \mathbf{w}_{dn}\} - \xi\{d_{n+1}, \mathbf{w}_{d(n+1)}\}. \quad (3.47)$$

The quantity $\Delta\xi_n$ approaches zero when the partial derivatives of condition (3.36) approach zero. A stronger statement that may sometimes hold is that “ $\Delta\xi_n$ approaches zero *only* when the partial derivatives approach zero” [37]. Mathematically, this statement means that for every $\epsilon > 0$, a $\delta > 0$ can be found such that

$$|\Delta\xi_n| \geq \delta \quad (3.48)$$

if

$$\left[\frac{\partial}{\partial d} \xi\{d_n, \mathbf{w}_{dn}\} \right]^2 + [\nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\}]^H [\nabla_{\mathbf{w}_{dn}} \xi\{d_n, \mathbf{w}_{dn}\}] \geq \epsilon. \quad (3.49)$$

The following lemma is stated and proven in [37].

Lemma. *If $\Delta\xi_n > 0$ for all n and $\Delta\xi_n$ approaches zero only when the partial derivatives approach zero, ξ_n must converge to a stationary point.* ■

This lemma provides a mean for determining the adaptation factors μ_n and α_n , since the MSE will converge to a stationary point if the adaptation factors can be determined such that $\Delta\xi_n > 0$ for all n and $\Delta\xi_n$ approaches zero only when the partial derivatives approach zero. The next proposition establishes the gain factors range for the above lemma to be true. It is an adaptation of proposition 2 of [37].

Proposition 3.1. Let the set of positive integers be divided arbitrarily into two disjoint subsets κ_1 and κ_2 , each containing an infinite number of positive integers. Let $\alpha_n = 0$ when $n \in \kappa_1$, and $\mu_n = 0$ when $n \in \kappa_2$. Let $\lambda_{\max}(n)$ be the maximum eigenvalue of the signal autocorrelation matrix \mathbf{R}_n and ϑ_n , the delay value closest to d_n , for which $\xi\{d_n, \mathbf{w}_{d_n}\}$ is minimum. The MSE will converge to a stationary point if

$$0 < \mu_n < \frac{1}{\lambda_{\max}(n)} \sin \left[\frac{\pi}{2(2\lceil d_n \delta \rceil + 1)} \right], \quad (3.50)$$

for $n \in \kappa_1$, and

$$0 < \alpha_n < 2 \left[\frac{\partial^2}{\partial d_n^2} \xi\{\vartheta_n, \mathbf{w}_{d_n}\} \right]^{-1}, \quad (3.51)$$

for $n \in \kappa_2$. In (3.50), the constant δ is $1/T$ in the case of a Type II-DAB adaptive system, and zero otherwise. The notation $\lceil d_n \delta \rceil$ means “the closest integer larger than $d_n \delta$ ”. ■

Proof: Consider first the condition $n \in \kappa_1$. In that case, $\alpha_n = 0$ and $d_{n+1} = d_n$. This situation corresponds to the usual adaptive filter convergence case, in which the MSE function ξ_n is a quadratic surface in the weight vector subspace, with a unique minimum with respect to \mathbf{w}_{d_n} . Then, equation (3.50) with $\delta = 0$ is the usual condition for convergence, at iteration n , of the adaptive transversal filter using the SD adaptation algorithm [6]. In the case of a Type II adaptive branch with the delay after the adaptive filter, (3.50) with $\delta = 1/T$ is the stability condition for integer delayed adjustments [49]. Since the performance surface is quadratic, $\Delta\xi_n$ approaches zero only when the gradient w.r.t. \mathbf{w}_n approaches zero, and the lemma ensures that a stationary point is reached. Note that if $n \in \kappa_1$ for M consecutive iterations, where M is the adaptive filter order, the autocorrelation matrix and its eigenvalues become time-independent in a Type I adaptive system.

In the other situation where $n \in \kappa_2$, $\mu_n = 0$ and the adaptive filter stays stationary. Then, from (3.5), and for a stationary reference filter

$$\Delta\xi_n = \mathbf{w}_{d_n}^H (\mathbf{R}_n - \mathbf{R}_{n+1}) \mathbf{w}_{d_n} - 2\text{Re}[\mathbf{w}_{d_n}(\mathbf{p}_n - \mathbf{p}_{n+1})] \quad (3.52)$$

and the variations of ξ_n w.r.t. d_n are function of both the autocorrelation $\phi_{yy}(n, n)$ of the adaptive filter output and of the cross-correlation function $\phi_{yr}(n, n)$ between this output and the reference signal. This function is generally multimodal w.r.t. d_n (see Section 3.2.1). It is therefore difficult to give a very precise idea of the delay tracking algorithm without knowing the actual value of d_n . Assuming a Taylor expansion of ξ_n around ϑ_n , the minimum closest to the actual value of d_n , the MSE function evaluated at $d_n = \vartheta_n$ is constant and the restricted class analysis of section 3.2.2.1 holds. Then (3.51) results from (3.28), with

$\gamma = \xi\{\vartheta_n, \mathbf{w}_{dn}\}$, and $\Delta\xi_n$ approaches zero only when the derivative of the MSE function w.r.t. d_n approaches zero. ■

This proposition states that, for any Type I or Type II structures, d_n and \mathbf{w}_n may be adjusted in any alternating fashion, and the MSE will converge to a stationary point if μ_n satisfies (3.50) during the adjustment of \mathbf{w}_n , and α_n satisfies (3.51) during the adjustment of d_n . The above condition is important because it confirms that, with the right parameters used *in alternation*, the MSE is reduced at each iteration and the joint SD algorithm converges eventually to a stationary point. Therefore, the algorithms of (3.34) and (3.35) can be used to track the variations of the reference system, if conditions (3.50) and (3.51) are satisfied. As for the algorithm of (3.33), the conditions of the theorem do not insure convergence, but they constitute a reference point for the selection of the proper adaptation constants.

3.3.1.3 Excess Mean Squared Error

The minimum MSE, given a certain value of d_n , was defined in (3.39) as $\xi_o(d_n)$. Denote the absolute MMSE as ξ_{\min} and define it as

$$\begin{aligned}\xi_{\min} &= \xi_o(D_n) \\ &= \phi_{rr}(0) - \mathbf{p}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n),\end{aligned}\tag{3.53}$$

where $\mathbf{p}(D_n)$ is the cross-correlation vector evaluated for $d_n = D_n$. Therefore, ξ_{\min} is the MSE for perfect cancellation or equalization by the joint adaptive structure. In steady-state conditions, any divergence from this perfect behaviour gives a MSE function greater or equal to ξ_{\min} .

The $(j+1)^{th}$ component of \mathbf{p}_n is given by $\phi_{ur}(-jT \pm d_n)$. Assuming steady-state conditions, $\phi_{ur}(-jT \pm d_n)$ can be approximated closely by the first three terms of its Taylor series expansion around the value $d_n = D_n$, i.e.

$$\phi_{ur}(-jT \pm d_n) \approx \phi_{ur}(-jT \pm D_n) + (d_n - D_n)\dot{\phi}_{ur}(-jT \pm D_n) + \frac{(d_n - D_n)^2}{2}\ddot{\phi}_{ur}(-jT \pm D_n),\tag{3.54}$$

where the dot denotes, as usual, the derivative with respect to d_n . Then, expressing it as a function of d_n , the cross-correlation vector can be approximated as

$$\mathbf{p}(d_n) \approx \mathbf{p}(D_n) + (d_n - D_n)\dot{\mathbf{p}}(D_n) + \frac{(d_n - D_n)^2}{2}\ddot{\mathbf{p}}(D_n).\tag{3.55}$$

Using (3.55) in (3.39) gives

$$\begin{aligned}
\xi_o(d_n) = & \phi_{rr}(0) - \mathbf{p}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n) \\
& - (d_n - D_n)^2 \left[\text{Re}[\ddot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n)] + \dot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\dot{\mathbf{p}}(D_n) \right] \\
& - 2(d_n - D_n)\text{Re}[\dot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n)] + (d_n - D_n)^3 \text{Re}[\ddot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n)] \\
& - 1/4(d_n - D_n)^4 \ddot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\ddot{\mathbf{p}}(D_n).
\end{aligned} \tag{3.56}$$

Assuming that d_n is close to D_n , the last two terms of (3.56) can be neglected. Furthermore, the expression $-2\text{Re}[\dot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n)]$ represents the derivative of $\xi_o(d_n)$ evaluated at its minimum, which is zero. Therefore $\xi_o(d_n)$ is approximately given by

$$\begin{aligned}
\xi_o(d_n) \approx & \phi_{rr}(0) - \mathbf{p}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n) \\
& - (d_n - D_n)^2 \left[\text{Re}[\ddot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\mathbf{p}(D_n)] + \dot{\mathbf{p}}^H(D_n)\mathbf{R}_n^{-1}\dot{\mathbf{p}}(D_n) \right],
\end{aligned} \tag{3.57}$$

and the excess MSE, defined as

$$\xi_{\text{ex}} = \xi_o(d_n) - \xi_{\text{min}}, \tag{3.58}$$

is

$$\xi_{\text{ex}} \approx -1/2(d_n - D_n)^2 \frac{\partial^2[\mathbf{p}^H(d_n)\mathbf{R}_n^{-1}\mathbf{p}(d_n)]}{\partial d_n^2} \Big|_{d_n=D_n}. \tag{3.59}$$

Note that from (3.53),

$$\frac{\partial^2[\mathbf{p}^H(d_n)\mathbf{R}_n^{-1}\mathbf{p}(d_n)]}{\partial d_n^2} \Big|_{d_n=D_n} = -\ddot{\xi}_{\text{min}}. \tag{3.60}$$

Combining the results of (3.59) and (3.60), the excess MSE is

$$\xi_{\text{ex}} \approx 1/2(d_n - D_n)^2 \ddot{\xi}_{\text{min}}. \tag{3.61}$$

Note that if the joint algorithm has converged near a local solution $d_n = \vartheta_n$, then the excess MSE *from that local minimum* is given by

$$\xi_{\text{ex}}^\vartheta \approx 1/2(d_n - \vartheta_n)^2 \ddot{\xi}\{\vartheta_n, \mathbf{w}_{\text{opt}}^\vartheta\}, \tag{3.62}$$

where

$$\mathbf{w}_{\text{opt}}^\vartheta = \mathbf{R}_n^{-1}\mathbf{p}(\vartheta_n). \tag{3.63}$$

The possibility of an excess MSE can be explained heuristically in the following way. For a finite-length adaptive filter of order M , the weight vector subspace is of dimension M . The delay subspace is always one-dimensional, irrespective of the value M . The adaptive filter attempts to model a time delay by shifting in time its weights by a corresponding amount. In

order to perform this operation without MSE increase, the weight vector subspace dimension has also to be increased by the same amount. If it is not, the adaptive filter algorithm seeks a compromise, within the fixed weight vector subspace, between reference filter modelling and delay modelling. This vector space view shows the inefficiency of the adaptive filter, in term of delay modelling, since the filter attempts to model a one-dimensional parameter (the delay) with a multi-dimensional component (the time shift in the weight vector).

3.3.1.4 Discussion

The convergence of the joint SD algorithm is not easy to characterize. By specializing the study to two special classes of adaptive systems, the convergence can be studied in qualitative terms. In these classes of systems, the only delay-dependent term is the cross-correlation function given by $\phi_{yr}(n, n) = \mathbf{w}_n^H \mathbf{p}_n$. The joint algorithm is then transformed to the one of (3.45). In this case the MSE function, as expressed in the adaptive weight vector subspace, is constant in shape (because the autocorrelation matrix is constant). The joint adaptive algorithm is such that the instantaneous MSE moves on the surface of the “bowl-shaped” MSE function, according to the adaptive weight vector, and the minimum of this bowl is modified, according to the delay d_n (since \mathbf{p}_n is function of this delay). The adaptive process converges when the first equation of (3.37) is verified.

As for the condition of convergence of the joint alternate algorithm given in Proposition 3.1, it provides some indications about the parameters that play a role in the joint algorithm convergence. In particular, if the MSE is close to its global minimum, the convergence bound for α_n is $2/\ddot{\xi}_{\min}$. This second derivative influences also the excess MSE, as shown in the previous subsection.

3.3.2 The Delay Tracking Properties of the Algorithm

The delay tracking properties of the joint SD algorithm are specifically studied in this subsection, with a special attention given to the cancellation and equalization structures in both Type I and Type II mode. The MSE function, for these configurations, is first considered. Then the SD delay tracking algorithm, as a constituent of the joint SD algorithm, is studied in details. The tracking mode assumption implies that both the reference filter $h(n)$ and the reference delay D_n are varying slowly.

3.3.2.1 The MSE Function for Specific Structures

In order to specify the MSE function for specific structures, the expressions of (3.12) to (3.14) are used with the proper value for $u(n)$ and $r(n)$ defined as in Figs. 2.6 and 2.7.

Therefore, $u(n) = y_1(n)$ and $r(n) = y_2(n)$ in cancellation configuration, while the inverse is true for the equalization configuration. Using the mathematical model of (3.1) and recalling that the noise processes are uncorrelated with every other signal, the following expressions for ξ_n in cancellation configuration are obtained.

Cancellation Configuration-Type I (delay before filter)

$$\begin{aligned} \xi_n^{(C)} = & \phi_{y_2 y_2}(n, n) + \sum_i \sum_j w_{ni}^* w_{nj} \phi_{y_1 y_1}(jT - iT + d_{n-j} - d_{n-i}) \\ & - 2\text{Re}[\sum_i \sum_j w_{ni}^* h^*(j) \phi_{ss}(jT - iT + D_{n-j} - d_{n-i})]. \end{aligned} \quad (3.64)$$

Cancellation Configuration-Type II-DAB (delay after filter in adaptive branch)

$$\begin{aligned} \xi_n^{(C)} = & \phi_{y_2 y_2}(n, n) + \sum_i \sum_j w_{(nT-d_n)i}^* w_{(nT-d_n)j} \phi_{y_1 y_1}(jT - iT) \\ & - 2\text{Re}[\sum_i \sum_j w_{(nT-d_n)i}^* h^*(j) \phi_{ss}(jT - iT + D_n - d_n)]. \end{aligned} \quad (3.65)$$

Cancellation Configuration-Type II-DRB (delay after filter in reference branch)

For the *alternate* Type II structure of Fig. 3.2, the MSE function is

$$\begin{aligned} \xi_n^{(C)} = & \phi_{y_2 y_2}(nT - d_n, nT - d_n) + \sum_i \sum_j w_{ni}^* w_{nj} \phi_{y_1 y_1}(jT - iT) \\ & - 2\text{Re}[\sum_i \sum_j w_{ni}^* h^*(j) \phi_{ss}(jT - iT + D_n + d_n)]. \end{aligned} \quad (3.66)$$

Similar expressions are obtained for the equalization configurations.

In tracking mode, it is assumed that the adaptive filter has fully adapted to the characteristics of $h(n)$ and is at least as long as the impulse response $h(n)$. For *high signal-to-noise ratios*, the i^{th} adaptive filter coefficients w_{ni} , at iteration n , is approximately of the form

$$w_{ni}^* \approx \begin{cases} h(i) & \text{System identification (cancellation)} \\ h^{-1}(i) & \text{Inverse filtering (equalization),} \end{cases} \quad (3.67)$$

where $h(i)$ is the i^{th} weight of the reference path filter, and is constant. In delay tracking mode, the only part of ξ_n that is of importance is the delay-dependent one. Define this

quantity as ν_n . Then, from (3.64) to (3.66),

$$\begin{aligned} \nu_n^{(C,I)} &= \sum_i \sum_j h(i)h(j)^* \phi_{y_1 y_1}(jT - iT + d_{n-j} - d_{n-i}) \\ &\quad - 2\text{Re} \left[\sum_i \sum_j h(i)h^*(j) \phi_{ss}(jT - iT + D_{n-j} - d_{n-i}) \right] \quad \text{Type I} \end{aligned} \quad (3.68)$$

$$\nu_n^{(C,II)} = -2\text{Re} \left[\sum_\ell \rho_h(\ell) \phi_{ss}(-\ell T + D_n \pm d_n) \right] \quad \text{Type II} \quad (3.69)$$

$$\nu_n^{(E,I)} = -2\text{Re} \left[\sum_i \sum_j h^{-1}(i)h(j) \phi_{ss}(-jT - iT - D_{n-j} \pm d_n) \right] \quad \text{Type I} \quad (3.70)$$

$$\begin{aligned} \nu_n^{(E,II)} &= \sum_i \sum_j h^{-1}(i)h^{-1*}(j) \phi_{y_2 y_2}(jT - iT + d_{n-j} - d_{n-i}) \\ &\quad - 2\text{Re} \left[\sum_i \sum_j h^{-1}(i)h(j) \phi_{ss}(-jT - iT - D_n - d_{n-i}) \right] \quad \text{Type II,} \end{aligned} \quad (3.71)$$

where $\rho_h(k)$ is the deterministic autocorrelation of the reference filter impulse response and is defined as

$$\rho_h(k) = \sum_i h(k+i)h^*(i). \quad (3.72)$$

Note that in expressions (3.69) and (3.70), the plus sign in front of d_n applies when the adaptive delay is transferred in the reference branch.

It is interesting to compare the above delay-dependent terms, especially when it is assumed that the reference delay D_n varies slowly. In this case, it can be assumed that both D_n and d_n are approximately constant over M samples (the filter time span), i.e. in both the reference and the adaptive filter delay lines, all the samples are approximately influenced by a constant delay. Then, the type of structure does not affect ν_n , which is now of the form

$$\nu_n^{(C)} \approx -2\text{Re} \left[\sum_\ell \rho_h(\ell) \phi_{ss}(-\ell T + D_n \pm d_n) \right] \quad (3.73)$$

$$\nu_n^{(E)} \approx -2\text{Re}[\phi_{ss}(D_n \mp d_n)]. \quad (3.74)$$

Comparing (3.73) and (3.74), it is noticed that the cancellation configuration is influenced by the form of both the deterministic autocorrelation $\rho_h(n)$ and the input signal autocorrelation $\phi_{ss}(\tau)$, while the equalization configuration is a function of only $\phi_{ss}(\tau)$. Since $\phi_{ss}(\tau)$ exhibits a maximum at $\tau = 0$, $\nu_n^{(E)}$ has a global minimum at $d_n = \pm D_n$. In the cancellation scenario, the characteristics of the delay tracking loop are functions of the reference filter

$h(n)$, but because of the definition of $\rho_h(n)$, there is a single global minimum corresponding to $d_n = \mp D_n$ ($\rho_h(n)$ has a maximum at $n = 0$). The two expressions of (3.73) and (3.74) are used next to characterize the SD descent delay tracking algorithm.

3.3.2.2 The SD Delay Tracking Algorithm

The results obtained in Subsection 3.2.2 are utilized in the following, in order to analyse the delay tracking portion of the joint SD algorithm. Therefore, $\gamma_n(d_n) = \xi_n$ and $\Theta_n = D_n$, for the cancellation configuration and $\Theta_n = -D_n$, for the equalization structure. It is also assumed that the adaptive filter has fully adapted to the time-invariant reference impulse response $h(n)$, and that $d_n = \pm D_n$. Because of this assumption, the error is minimum and the corresponding MSE is equal to the MMSE ξ_{\min} (see Subsection 3.3.1.3). Then $\ddot{\xi}_n = \ddot{\xi}_{\min}$ and is constant with time, which allows the use of the results of Subsection 3.2.2.1. Making use of (3.28) and (3.29), the stability range for α is

$$0 < \alpha < 2/\ddot{\xi}_{\min}, \quad (3.75)$$

and the time constant of delay adaptation is

$$\tau_{\text{del}} \approx \frac{1}{\alpha \ddot{\xi}_{\min}}. \quad (3.76)$$

Tighter or more explicit bounds for α can be easily obtained for particular cases.

Bounds in High Signal-to-Noise Ratios Conditions

The derivative of a bandlimited continuous-time signal can be obtained from the samples of that signal by using a wideband differentiator with frequency response given by [50]

$$H_{\text{diff}}(e^{j\omega}) = j\omega/T \quad -\pi < \omega < \pi. \quad (3.77)$$

Then, for the cancellation configuration, $\ddot{\phi}_{yr}^{(C)}(n, n)$ can be expressed in the frequency domain, with $d_n = d$ and $D_n = D$, as (see equation (3.73))

$$\ddot{\phi}_{yr}^{(C)}(n, n) \approx -\frac{1}{2\pi T^2} \int_{-\pi}^{\pi} \omega^2 |H(e^{j\omega})|^2 e^{-j\omega(d \pm D)/T} \Phi_{ss}(e^{j\omega}) d\omega, \quad (3.78)$$

where $H(e^{j\omega})$ is the transfer function of the reference filter and $\Phi_{ss}(e^{j\omega})$ is the power spectral density of the signal $s(n)$.

Defining the maximum value of the input signal power spectral density $\Phi_{ss}(e^{j\omega})$ as Φ_{\max} , the cross-correlation function is, when $d = \mp D$,

$$\ddot{\phi}_{yr}^{(C)}(n, n) \leq \frac{\Phi_{\max}}{2\pi T^2} \int_{-\pi}^{\pi} -\omega^2 |H(e^{j\omega})|^2 d\omega. \quad (3.79)$$

But

$$-\frac{1}{2\pi T^2} \int_{-\pi}^{\pi} \omega^2 |H(e^{j\omega})|^2 d\omega = \rho_h''(0), \quad (3.80)$$

where the prime denotes the derivative with respect to the *continuous-time* correlation argument. Then

$$\ddot{\phi}_{yr}^{(C)}(n, n) \leq \Phi_{\max} \rho_h''(0). \quad (3.81)$$

Noting that, for slowly varying delays, $\ddot{\xi}_{\min} = -2\text{Re}[\ddot{\phi}_{yr}^{(C)}(n, n)]$, (3.75) becomes

$$0 < \alpha < \frac{-1}{\Phi_{\max} \text{Re}[\rho_h''(0)]} \quad \text{Cancellation.}$$

Using the same type of development for equalization, (3.75) becomes

$$0 < \alpha < \frac{3T^2}{\Phi_{\max} \pi^2} \quad \text{Equalization.}$$

The following proposition has therefore been established.

Proposition 3.2. *In tracking mode and in high signal-to-noise ratios conditions, a sufficient range of convergence for the delay gain factor is*

$$0 < \alpha < \frac{-1}{\Phi_{\max} \text{Re}[\rho_h''(0)]} \quad (3.82)$$

for the cancellation configuration and

$$0 < \alpha < \frac{3T^2}{\Phi_{\max} \pi^2} \quad (3.83)$$

for the equalization configuration. ■

Bounds for White Signals

Assume that the input signal and the noise signals are white with respective power spectral densities Φ_{ss} , $\sigma_{v_1}^2$ and $\sigma_{v_2}^2$. Then, it can be shown that the optimum impulse response of the adaptive filter, in steady-state conditions, is

$$w_{\text{opt}}(n) = \begin{cases} \frac{\Phi_{ss}}{\Phi_{ss} + \sigma_{v_1}^2} h^*(n) & \text{Cancellation} \\ \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{H(e^{-j\omega}) \Phi_{ss}}{|H(e^{-j\omega})|^2 \Phi_{ss} + \sigma_{v_1}^2} e^{j\omega n} d\omega & \text{Equalization,} \end{cases} \quad (3.84)$$

where a noncausal system, with an infinite impulse response, is assumed for the equalization case. Then, using a development analog to the high SNR one, the following double derivative for the cross-correlation functions are of the form

$$\ddot{\phi}_{yr}^{(C)}(n, n) = \frac{\Phi_{ss}^2}{\Phi_{ss} + \sigma_{v_1}^2} \rho_h''(0) \quad (3.85)$$

$$\ddot{\phi}_{yr}^{(E)}(n, n) = \frac{-\Phi_{ss}^2}{2\pi T^2} \int_{-\pi}^{\pi} \omega^2 \frac{|H(e^{j\omega})|^2}{|H(e^{j\omega})|^2 \Phi_{ss} + \sigma_{v_1}^2} d\omega. \quad (3.86)$$

The bound of (3.75) can then be written with $\ddot{\xi}_{\min} = -2\text{Re}[\ddot{\phi}_{yr}(n, n)]$. Note that these bounds for white signals reduce to the previous ones if the signal-to-noise ratios are high.

Relation Between the Adaptive Delay and the Adaptive Filtering Processes

In general, it is desired that the compensation for the reference delay, in the adaptive branch, be performed by the adaptive delay element alone. Since the adaptive filter can potentially adapt to the reference delay, the time constant of adaptation of the delay element should be smaller than the time constant for the adaptive filter. The time constant τ_j of the j^{th} mode of adaptation of the *normalized* adaptive filter weight vector is [7],

$$\tau_j \approx \frac{1}{2\mu\lambda_j}, \quad (3.87)$$

where λ_j is the j^{th} eigenvalue of the input signal autocorrelation matrix \mathbf{R} .

The adaptation time constant of the j^{th} mode of the MSE function, as a function of the adaptive weight vector, is

$$(\tau_{\text{mse}})_j \approx \frac{\tau_j}{2} = \frac{1}{4\mu\lambda_j}, \quad (3.88)$$

i.e. the MSE function converges twice as fast as the adaptive weight vector when the delay element is assumed fixed. The fastest influence of the adaptive filter on the MSE curve therefore has the time constant

$$(\tau_{\text{mse}})_{\text{min}} \approx \frac{1}{4\mu\lambda_{\text{max}}}. \quad (3.89)$$

A speed of convergence constraint can be applied on the adaptive filter, in order to restrict the influence of any reference delay variations on its behaviour, i.e. the condition that the adaptive delay time constant should be much smaller than $(\tau_{\text{mse}})_{\text{min}}$ is imposed. Assuming that the adaptive delay element settles down after 5 time constants, an upper bound on τ_{del} is

$$\tau_{\text{del}} < \frac{(\tau_{\text{mse}})_{\text{min}}}{5}$$

or, using equation (3.89),

$$\tau_{\text{del}} < \frac{1}{20\mu\lambda_{\text{max}}}. \quad (3.90)$$

Equations (3.76) and (3.90) give a lower bound on the delay adaptation factor α , i.e.

$$\frac{20\mu\lambda_{\text{max}}}{\xi_{\text{min}}} < \alpha. \quad (3.91)$$

This gives a relation between the adaptation factors, μ and α , of the coupled processes, when the constraint is applied. Equations (3.82) and (3.83) can be combined with (3.91) in order to obtain the following proposition.

Proposition 3.3. *Assuming that the adaptive delay element has a time constant five times smaller than the time constant of the fastest adaptive filter mode of adaptation, then the delay element gain factor satisfies the following conditions*

$$\frac{20\mu\lambda_{\max}}{\xi_{\min}} < \alpha < \frac{-1}{\Phi_{\max}\text{Re}[\rho_h''(0)]} \quad \text{Cancellation} \quad (3.92)$$

and

$$\frac{20\mu\lambda_{\max}}{\xi_{\min}} < \alpha < \frac{3T^2}{\Phi_{\max}\pi^2} \quad \text{Equalization.} \quad (3.93)$$

■

Adaptive Delay Response to a Reference Delay Step

The use of the linearized version of the adaptive delay equation (equation (3.25), with $\gamma_n = \xi_n$) assumes implicitly that the main lobe of the MSE function (the main lobe is defined as the region between the two inflexion points, U_1 and U_2 , located on each side of the global maximum at $d_n = D_n$) can be closely approximated by a quadratic function of d_n (i.e. the higher terms in (3.21) are neglected). In addition to the fact that this approximation becomes worse as d_n gets further away from D_n , it is also limited by the width of the main lobe of the MSE function. The main lobe has a width defined as $|U_1 - U_2|$ and, assuming that the adaptive delay element tracks perfectly D_n , the maximum allowable input delay step is

$$\Delta_{\max} = \frac{|U_1 - U_2|}{2}, \quad (3.94)$$

since, for slowly varying delays, the MSE function is symmetric with respect to D_n (see equations (3.73) and (3.74)). If Δ_{\max} is larger than the main lobe width, the adaptive delay is likely to converge to a local minimum of the MSE function. In general, the main lobe width is a function of both the reference filter and the input signal autocorrelation function, as shown in (3.73). Assume that $\Phi_{ss}(e^{j\omega})$ is white with unit variance and that the reference filter is an ideal lowpass filter, i.e.

$$H(e^{j\omega}) = \begin{cases} 1 & -\pi \leq \omega \leq \pi \\ 0 & \text{otherwise.} \end{cases} \quad (3.95)$$

Then, from (3.73), the delay-dependent part of the MSE function is

$$\begin{aligned} \nu &= \frac{-1}{\pi} \int_{-\pi}^{\pi} e^{-j\omega(d-D)/T} d\omega \\ &= -2T \frac{\sin \pi(D-d)/T}{\pi(D-d)} \end{aligned} \quad (3.96)$$

for which the main lobe is symmetric and approximately 2 samples wide. For a coloured input and a non-flat filter, the main lobe is likely to be of larger width, and the following proposition has been established.

Proposition 3.4. *For any type of configuration in tracking mode and for slowly varying delays, a conservative upper bound on the maximum allowable instantaneous reference delay deviation is on the order of one sample (or T seconds).* ■

In order to see the effect of a delay step on the adaptive delay, assume that at iteration $n = 0$, a constant delay of Δ samples is applied in excess of D_0 , i.e. the reference delay D_n is

$$D_n = D_0 + \Delta$$

for $0 < n$. Assume also that Δ is lower than one, and that (3.92) or (3.93) are satisfied. The adaptive delay value, in excess of D_0 , is given in (3.27), with $\ddot{\gamma} = \ddot{\xi}_{\min}$ and $\Theta_{n-\ell-1} = \Delta$, i.e.

$$d_n = (1 - \alpha \ddot{\xi}_{\min})^n D_0 + \alpha \Delta \ddot{\xi}_{\min} \sum_{\ell=0}^{n-1} (1 - \alpha \ddot{\xi}_{\min})^\ell. \quad (3.97)$$

Equation (3.97) can be written as

$$\begin{aligned} d_n &= (1 - \alpha \ddot{\xi}_{\min})^n D_0 + \alpha \Delta \ddot{\xi}_{\min} \frac{1 - (1 - \alpha \ddot{\xi}_{\min})^n}{\alpha \ddot{\xi}_{\min}} \\ &= (1 - \alpha \ddot{\xi}_{\min})^n (D_0 - \Delta) + \Delta, \end{aligned} \quad (3.98)$$

which, if the algorithm converges, tends toward $d_n = \Delta$ when $n \rightarrow \infty$.

In summary, the response of the joint SD algorithm with linearized delay equation, when the time constant of delay adaptation is much smaller than the time constant of the filter adaptation, is such that the delay element compensates completely for the delay step, after a transient period.

Adaptive Delay Response to a Reference Delay Ramp

The reference delay is assumed to be of the form

$$D_n = D_0 + An \quad 0 < n,$$

where A is the slope of the delay ramp, in samples/sample.

Assuming that conditions (3.92) or (3.93) are satisfied, an analysis similar to the one given for the delay step shows that, after a transient period, the delay element value is

$$d_{ss} = An - A - \frac{A(1 - \alpha \ddot{\xi}_{\min})}{\alpha \ddot{\xi}_{\min}} \quad (3.99)$$

and lags the input by

$$\text{lag} = \frac{A}{\alpha \ddot{\xi}_{\min}}. \quad (3.100)$$

Using (3.76) in (3.100) gives

$$\text{lag} \approx A\tau_{\text{del}}. \quad (3.101)$$

This lag error constitutes the residual error that the delay element cannot cope with. It appears as a *constant* delay at the input of the adaptive filter and can therefore be compensated for by the filter, after a transient period [†].

The maximum allowable slope in the input delay is dictated by the width of the main lobe of the MSE function. In order to allow continuous tracking of the input, the delay element lag error must be smaller than the maximum allowable input delay step, defined in (3.94), i.e.

$$\text{lag} < \Delta_{\text{max}}$$

or

$$\frac{A}{\alpha\ddot{\xi}_{\text{min}}} < \frac{|U_1 - U_2|}{2}. \quad (3.102)$$

The slope of the input ramp must also be such that the delay change occurring over one sample is less than Δ_{max} , i.e. such that

$$A < \frac{|U_1 - U_2|}{2}. \quad (3.103)$$

The following proposition is then established.

Proposition 3.5. *An upper bound on the maximum allowable input slope is*

$$A < \begin{cases} \alpha\ddot{\xi}_{\text{min}} \frac{|U_1 - U_2|}{2} & \text{if } \alpha\ddot{\xi}_{\text{min}} < 1 \\ \frac{|U_1 - U_2|}{2} & \text{if } 1 \leq \alpha\ddot{\xi}_{\text{min}}. \end{cases} \quad (3.104)$$

From the conservative upper bound derived in Proposition 3.4, a conservative upper bound on the slope is

$$A < \begin{cases} \alpha\ddot{\xi}_{\text{min}} & \text{if } \alpha\ddot{\xi}_{\text{min}} < 1 \\ 1 & \text{if } 1 \leq \alpha\ddot{\xi}_{\text{min}}. \end{cases} \quad (3.105)$$

■

Note that these bounds can be very loose. This is so because they make use of the maximum allowable input *step* (Δ_{max}) to bound the input *change* over one sample. Since the adaptive delay algorithm does not allow for a perfect correction in a one sample time, further input change by Δ_{max} will bring the adaptive loop out of its tracking range. Therefore, the bound of (3.104) can correspond sometimes only to a gross indication of the value of the input slope.

[†] Note that if the adaptive delay element were not present, the adaptive filter would face a nonstationary delay, which would produce an excess MSE that increases with time. The combination of the adaptive delay and the adaptive filter results in a fixed excess MSE.

Adaptive Delay Response to a Sinusoidally Changing Reference Delay

Assume that the reference delay is of the form

$$D_n = D_0 + A \sin(2\pi n/P) \quad 0 < n,$$

where A is the waveform amplitude and P is its period, both expressed in samples. Assume again that conditions (3.92) or (3.93) are satisfied.

With $D_n = A \sin(2\pi n/P)$, the linearized delay equation becomes

$$d_{n+1} = (1 - \alpha \ddot{\xi}_{\min})d_n + \alpha A \ddot{\xi}_{\min} \sin(2\pi n/P). \quad (3.106)$$

After some manipulations, the solution is

$$d_n = \alpha A \ddot{\xi}_{\min} \sin(2\pi/P) \{2|K_1| \cos[2\pi(n-1)/P + \theta] + K_2(1 - \alpha \ddot{\xi}_{\min})^{n-1}\} U_s(n-1) \quad (3.107)$$

where

$$K_1 = \frac{e^{j2\pi/P}}{2j \sin(2\pi/P)(e^{j2\pi/P} - (1 - \alpha \ddot{\xi}_{\min}))} \quad (3.108)$$

$$K_2 = \frac{(1 - \alpha \ddot{\xi}_{\min})}{(1 - \alpha \ddot{\xi}_{\min})^2 - 2(1 - \alpha \ddot{\xi}_{\min}) \cos 2\pi/P + 1}, \quad (3.109)$$

the variable θ is the phase of K_1 and $U_s(n)$ is the unit step function defined as

$$U_s(n) = \begin{cases} 0 & n < 0 \\ 1 & 0 \leq n. \end{cases} \quad (3.110)$$

Equation (3.107) shows that after a transition period, the steady-state delay is

$$d_{ss} = 2\alpha A \ddot{\xi}_{\min} \sin(2\pi/P) |K_1| \cos[2\pi(n-1)/P + \theta]. \quad (3.111)$$

Using (3.108) and (3.111), the magnitude of this sinusoidal waveform is

$$\begin{aligned} I &= 2\alpha A \ddot{\xi}_{\min} \sin(2\pi/P) |K_1| \\ &= \frac{\alpha \ddot{\xi}_{\min} A}{|e^{j2\pi/P} + \alpha \ddot{\xi}_{\min} - 1|}. \end{aligned} \quad (3.112)$$

If P is assumed large, compared to the time constant τ_{del} , the denominator of (3.112) is approximately equal to $\alpha \ddot{\xi}_{\min}$ and $I \approx A$. The steady-state delay solution is then

$$\begin{aligned} d_{ss} &\approx A \cos[2\pi(n-1)/P + \theta] \\ &\approx A \sin[2\pi n/P + \theta + \pi/2 - 2\pi/P]. \end{aligned} \quad (3.113)$$

Therefore, if (3.92) or (3.93) are satisfied, the delay element follows closely the reference delay, with a phase lag $(2\pi/P - \pi/2 - \theta)$ and a slightly smaller amplitude. The steady-state difference between D_n and d_n is sinusoidal and influences the adaptive filter behaviour. In steady-state, the adaptive filter coefficients therefore vary sinusoidally.

3.3.2.3 Discussion

Recall the main assumptions used in Subsection 3.3.2 for the study of the delay tracking properties of the joint SD algorithm:

1. The signal-to-noise ratios are often assumed high enough such that the adaptive filter Wiener solution is approximately equal to the reference filter $h(n)$ or its inverse.
2. The adaptive filter has fully adapted to the Wiener solution.
3. The delays D_n and d_n vary slowly such that the samples across any filter delay line are affected by the same delay.
4. The reference filter is time-invariant.
5. The second derivative of the MSE function, when evaluated at $d_n = D_n$, is constant and equal to ξ_{\min} .
6. The adaptive filter time constant of adaptation is much smaller than the adaptive delay time constant.

The first and second assumptions were essentially used to simplify the study of the delay tracking algorithm. The first one is not necessarily true in practice, but it simplifies the analysis and gives useful results. The second assumption is justified, since one is interested primarily in perfect delay tracking, which happens when the Wiener solution is attained. The results obtained using this assumption, essentially the restricted convergence ranges of Proposition 3.2, are therefore significantly useful in the application of the delay tracking algorithm. The third and fourth assumptions are also used for the sake of simplicity and are not necessarily true in practice. In particular, the function of the adaptive filter is to track the variations of the reference filter. When this happens, assumption 5 is hardly justified and assumption 6 limits the tracking ability of the adaptive filter. But when the reference filter variations are slow, compared to the reference delay variations, $h(n)$ is quasi-stationary over a limited period of time, and both assumptions 5 and 6 are justified. In fact, the last three assumptions are intimately linked, since practical considerations justify assumption 6, which itself supports assumptions 4 and 5.

In practice, the adaptive filter is expected to compensate for some of the reference delay variations. But the effect of these compensations, in the adaptive delay vector space, is to change the minimum location, without affecting significantly the second derivative of ξ_n at this minimum. The results obtained with assumption 5 are therefore of importance, since assumption 6 should be met in practice.

A major problem could nevertheless happen in the case of a practical finite-length adaptive filter. In this case, the joint algorithm could converge to a stationary solution for

which the MSE is higher than the optimum that could be achieved if the adaptive delay d_n models perfectly D_n (i.e. the excess MSE is nonzero). This could be so because, if $d_n \neq D_n$, the adaptive filter converges to a solution where the optimum weight vector is shifted, in order to compensate for delay difference, and modified to obtain the MMSE corresponding to this shift. If the adaptive filter is of infinite length and noncausal, all such solutions reach the same MMSE, but if the filter length is limited, so is its modelling capability and the MMSE is then at least as large as for the optimum solution. The difference between the MMSE and the actual MSE is the excess MSE, as defined in Subsection 3.3.1.3.

3.4 Application of the Joint SD Algorithm

In this section, the results derived for the SD delay tracking algorithm are specialized to some specific cases. The application of the algorithm, for the tracking of the reference branch variations, is governed mainly by four expressions. These equations are (3.50), (3.75), (3.76) and (3.61) and are reproduced next.

$$\begin{aligned} 0 < \mu < \frac{1}{\lambda_{\max}(n)} \sin \left[\frac{\pi}{2(2\lceil d_n \delta \rceil + 1)} \right] \\ 0 < \alpha < 2/\ddot{\xi}_{\min} \\ \tau_{\text{del}} &\approx \frac{1}{\alpha \ddot{\xi}_{\min}} \\ \xi_{\text{ex}} &\approx 1/2(d_n - D_n)^2 \ddot{\xi}_{\min}. \end{aligned}$$

3.4.1 The Function $\ddot{\xi}_{\min}$ in Cancellation Configuration

The function $\ddot{\xi}_{\min}$ is examined in some detail in this subsection, since it is used in every expression of importance in the SD delay tracking algorithm. The investigation is limited to *real signals and systems* in cancellation configuration.

From (3.73), (3.74) and (3.78), $\ddot{\xi}_{\min}$ is given by (for the high signal-to-noise ratio case) [†]

$$\ddot{\xi}_{\min} \approx \frac{1}{\pi T^2} \int_{-\pi}^{\pi} \omega^2 |H(e^{j\omega})|^2 \Phi_{ss}(e^{j\omega}) d\omega. \quad (3.114)$$

For white input and noise processes, $\ddot{\xi}_{\min}$ is of the form

$$\begin{aligned} \ddot{\xi}_{\min} &= \frac{\Phi_{ss}^2}{(\Phi_{ss} + \sigma_{v_1}^2) \pi T^2} \int_{-\pi}^{\pi} \omega^2 |H(e^{j\omega})|^2 d\omega \\ &= \frac{-2\Phi_{ss}^2 \rho_h''(0)}{\Phi_{ss} + \sigma_{v_1}^2}. \end{aligned} \quad (3.115)$$

[†] This equation applies to both the cancellation and the equalization scenarios. For the latter one, $H(e^{j\omega})$ is simply taken to be unity for $-\pi < \omega < \pi$.

If the reference branch signal-to-noise ratio is high enough, note that the integral of (3.114) is approximately equal to $-2\phi''_{rr}(0)$, where the prime denotes the derivative with respect to the correlation argument. In this case,

$$\ddot{\xi}_{\min} \approx \begin{cases} \frac{-2\Phi_{ss}}{\Phi_{ss} + \sigma_{v_1}^2} \phi''_{rr}(0) & \text{White processes} \\ -2\phi''_{rr}(0) & \text{High SNR's.} \end{cases} \quad (3.116)$$

The quantity $\ddot{\xi}_{\min}$ can be approximated by different numerical methods [51]. A simple, although not very reliable one, is obtained by differentiating twice Stirling's formula for polynomial approximation of the function $\phi_{rr}(\tau)$. This gives

$$\ddot{\xi}_{\min} \approx \frac{4\Phi_{ss}}{\Phi_{ss} + \sigma_{v_1}^2} \frac{\phi_{rr}(0) - \phi_{rr}(k)}{k^2}, \quad (3.117)$$

for a small constant k .

Bandlimited Reference filter

Assume that the reference filter is limited to the range $-\omega_1 \leq \omega \leq \omega_1$. Then, because the function ω^2 is positive and because of the real system assumption, (3.114) can be transformed, using the mean value theorem, to

$$\begin{aligned} \ddot{\xi}_{\min} &\approx \frac{1}{\pi T^2} \int_{-\omega_1}^{\omega_1} \omega^2 |H(e^{j\omega})|^2 \Phi_{ss}(e^{j\omega}) d\omega \\ &= \frac{2B\omega_1^2}{\pi T^2} \int_0^{\omega_1} |H(e^{j\omega})|^2 \Phi_{ss}(e^{j\omega}) d\omega, \end{aligned} \quad (3.118)$$

and (3.115) to

$$\begin{aligned} \ddot{\xi}_{\min} &= \frac{2B\omega_1^2 \Phi_{ss}^2}{(\Phi_{ss} + \sigma_{v_1}^2) \pi T^2} \int_0^{\omega_1} |H(e^{j\omega})|^2 d\omega \\ &= \frac{2\Phi_{ss}^2 B\omega_1^2}{(\Phi_{ss} + \sigma_{v_1}^2) T^2} \rho_h(0) \\ &\leq \frac{2\Phi_{ss}^2 \omega_1^2}{(\Phi_{ss} + \sigma_{v_1}^2) T^2} \rho_h(0), \end{aligned} \quad (3.119)$$

where B is a real constant between zero and one. For a *large reference branch signal-to-noise ratio*

$$\begin{aligned} \ddot{\xi}_{\min} &= \frac{2\Phi_{ss} B\omega_1^2}{(\Phi_{ss} + \sigma_{v_1}^2) T^2} \phi_{rr}(0) \\ &\leq \frac{2\Phi_{ss} \omega_1^2}{(\Phi_{ss} + \sigma_{v_1}^2) T^2} \phi_{rr}(0). \end{aligned} \quad (3.120)$$

Note that, from (3.115) and (3.119),

$$B = \frac{-T^2 \rho_h''(0)}{\omega_1^2 \rho_h(0)} \quad (3.121)$$

and for large reference branch signal-to-noise ratios

$$B \approx \frac{-T^2 \phi''_{rr}(0)}{\omega_1^2 \phi_{rr}(0)}. \quad (3.122)$$

3.4.2 Discussion

The results of Subsection 3.4.1 can be used in practice, for determining the gain factor α , the time constant τ_{del} and the excess MSE ξ_{ex} . For a high reference branch signal-to-noise ratio, $\ddot{\xi}_{\text{min}}$ can be approximated directly, using (3.117), by measuring the reference signal power and its autocorrelation at a small lag. The quantity can also be upper-bound by the value $2\Phi_{ss}\pi^2\phi_{rr}(0)/(\Phi_{ss} + \sigma_{v_1}^2)T^2$, obtained with $\omega_1 = \pi$ in (3.120). Equations (3.118) to (3.120) also show that $\ddot{\xi}_{\text{min}}$ is proportional to the reference filter bandwidth, to the input signal power and to $\rho_h(0)$. Therefore, these three parameters limit both the gain factor and the time constant, and increase the excess MSE. This is illustrated in Table 3.1, where \propto means *proportional to* and $(\alpha)^{-1}$ denotes *inversely proportional to*.

	α_{max}	τ_{del}	ξ_{ex}
Φ_{ss}	$(\alpha)^{-1}$	$(\alpha)^{-1}$	α
ω_1	$(\alpha)^{-1}$	$(\alpha)^{-1}$	α
$\rho_h(0)$	$(\alpha)^{-1}$	$(\alpha)^{-1}$	α

Table 3.1 Critical parameters in the joint SD algorithm

3.5 Summary

Joint time delay estimation and adaptive MMSE filtering, using the steepest-descent algorithm, has been studied in detail in this chapter. The MSE function was shown to be dependent on the form of the joint structure, and the evolution of the joint algorithm estimates was investigated qualitatively. The conditions of convergence of the joint SD algorithm were investigated, when the adaptive delay element and the adaptive filter are adapted alternatively. The excess MSE was derived, in order to express how well the joint algorithm tracks the optimum solution. When the reference delay is assumed to evolve

slowly, the adaptive delay adaptation factor and time constant are shown to be inversely proportional to the second derivative of the MMSE. Some bounds on the reference delay variations were derived, in order to allow proper delay tracking. Finally, some details were given about the practical application of the joint SD algorithm. The material presented in this chapter shows the possibilities and limitations of the joint time delay estimation and adaptive filtering algorithm based on the MMSE criterion, when a steepest-descent algorithm is used. It is useful in the design of more practical algorithms in which the gradient and derivative have to be estimated, and is of importance in the application of the joint LMS algorithm presented in the next chapter.

4.1 Introduction

In order to implement the joint steepest-descent algorithm presented in the previous chapter, the MSE gradient with respect to the adaptive weight vector and the MSE derivative with respect to the adaptive delay both have to be estimated. This can be accomplished in various ways, in particular by approximating the derivatives with difference equations [6], or by approximating the MSE function $\xi_n = E[|e(n)|^2]$ with the instantaneous squared error $\gamma_n = |e(n)|^2$, and by applying the SD algorithm. This last option corresponds to the least-mean-square (LMS) algorithm [10] and is the subject of this chapter.

Consider a cancellation configuration. In order to derive the LMS algorithm, rewrite the error in equation (3.2) as $e(n, d_n)$, where the dependence on the delay estimate is denoted explicitly. In a Type I adaptive system, it is assumed that the delay d_n propagates instantaneously into the adaptive filter delay line and the error can be expressed as

$$e(n, d_n) = r(n) - y(n, d_n) \quad \text{Type I,} \quad (4.1)$$

where the adaptive branch output $y(n, d_n)$ is defined as

$$y(n, d_n) = \mathbf{w}_n^H \mathbf{u}(nT - d_n), \quad (4.2)$$

and $\mathbf{u}(nT - d_n)$ is the delayed vector of input samples defined in equation (3.9). In a Type II structure, the adaptive delay can be located in either the adaptive branch or the reference

branch and the error can take two forms. If the delay element is in the adaptive branch, the error is defined as

$$e(n, d_n) = r(n) - y(nT - d_n) \quad \text{Type II-DAB,} \quad (4.3)$$

where

$$y(nT - d_n) = \mathbf{w}_{nT-d_n}^H \mathbf{u}(nT - d_n). \quad (4.4)$$

If the delay is in the reference branch, the error is

$$e(n, d_n) = r(nT - d_n) - y(n) \quad \text{Type II-DRB.} \quad (4.5)$$

In the adaptive weight vector subspace, it is well known that the LMS algorithm is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e^*(n, d_n) \mathbf{u}_n, \quad (4.6)$$

where \mathbf{u}_n is the vector of delayed input samples, defined in equations (3.8) or (3.9), and the error $e(n, d_n)$ is any of the errors in (4.1) to (4.5). In the adaptive delay subspace, the derivative estimate is given by

$$\hat{\nabla}_{d_n} \xi_n \triangleq \frac{\partial |e(n, d_n)|^2}{\partial d_n} = \begin{cases} -2\text{Re} \left[e^*(n, d_n) \frac{\partial y(n, d_n)}{\partial d_n} \right] & \text{Type I} \\ -2\text{Re} \left[e^*(n, d_n) \frac{\partial y(nT - d_n)}{\partial d_n} \right] & \text{Type II-DAB} \\ 2\text{Re} \left[e^*(n, d_n) \frac{\partial r(nT - d_n)}{\partial d_n} \right] & \text{Type II-DRB,} \end{cases} \quad (4.7)$$

corresponding to the three cases considered previously. The LMS adaptive delay algorithm is obtained by using the result of (4.7) in the SD adaptive delay algorithm, defined in equation (3.31).

The purpose of this chapter is to study the behaviour of the three forms of the joint LMS algorithm, defined by

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n + 2\mu e^*(n, d_n) \mathbf{u}_n \\ d_{n+1} &= d_n - \alpha \hat{\nabla}_{d_n} \xi_n, \end{aligned} \quad (4.8)$$

where (4.7) is used to define the derivative estimate. The only type of algorithm considered is the one corresponding to equation (3.33). In order not to obscure more than necessary the derivations, all signals and systems will be considered *real* in the analyses.

This chapter is mainly theoretical and addresses mostly the behaviour of the joint LMS algorithm in steady-state conditions. The convergence of the algorithm, from arbitrary

conditions, is considered first in Section 4.2. The analysis of the joint algorithm, in steady-state conditions, is performed in Section 4.3 for the Type I and the Type II (DAB and DRB) adaptive systems in cancellation configuration. The analyses presented in this section are for convergence in the mean and in the mean square, of both estimates d_n and w_n . The excess MSE and misadjustment are also considered for the three algorithms. A discussion of the results of Section 4.3 is then presented in Section 4.4 and their application in some special cases is considered in Section 4.5.

The main contributions of this chapter are the generalizations of LMS time delay estimation, and the extension of LMS adaptive filtering to the situation where the filter input signal and the reference signal experience different sampling rates. New results are derived about the convergence, in the mean and the mean square, of the two portions of the joint algorithm, as well as about the excess MSE and the misadjustment of the joint algorithm.

4.2 Convergence of the Joint LMS Algorithm Using the ODE Method

The convergence study of recursive stochastic algorithms is a difficult task and has been only partially successful. One type of algorithm has been analysed in some depth by Ljung [52] and is of the form

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \boldsymbol{\gamma}(n)\mathbf{R}^{-1}(n)\boldsymbol{\psi}(n)\epsilon(n), \quad (4.9)$$

where $\boldsymbol{\theta}(n)$ denotes the vector estimate at iteration n , $\boldsymbol{\gamma}(n)$ denotes a matrix gain sequence, $\boldsymbol{\psi}(n)$ is a regression vector (a data vector indicating a gradient search direction) and $\epsilon(n)$ represents an estimation error. The joint algorithm (4.9) is equivalent to (4.8) with the following definitions

$$\boldsymbol{\theta}(n) = \begin{bmatrix} \mathbf{w}_n \\ d_n \end{bmatrix} \quad (4.10)$$

$$\boldsymbol{\gamma}(n) = \begin{bmatrix} \mu_n & 0 \\ 0 & \alpha_n \end{bmatrix} \quad (4.11)$$

$$\mathbf{R}(n) = \mathbf{I} \quad (4.12)$$

$$\epsilon(n) = e(n, d_n). \quad (4.13)$$

and

$$\boldsymbol{\psi}(n) = \begin{cases} \begin{bmatrix} 2\mathbf{u}(nT - d_n) \\ 2\dot{y}(n, d_n) \end{bmatrix} & \text{Type I} \\ \begin{bmatrix} 2\mathbf{u}(nT - d_n) \\ 2\dot{y}(nT - d_n) \end{bmatrix} & \text{Type II-DAB} \\ \begin{bmatrix} 2\mathbf{u}(n) \\ -2\dot{r}(nT - d_n) \end{bmatrix} & \text{Type II-DRB.} \end{cases} \quad (4.14)$$

Note that it is explicitly assumed that the adaptation factors μ_n and α_n are function of time.

The matrix $\mathbf{R}(n)$ in (4.9) allows for the possibility of a Newton step, in which case $\mathbf{R}(n)$ is chosen as [53]

$$\mathbf{R}(n+1) = \mathbf{R}(n) + \gamma(n+1)[\boldsymbol{\psi}(n+1)\boldsymbol{\psi}^T(n+1) - \mathbf{R}(n)]. \quad (4.15)$$

Ljung proposes in [52] an approach that relies on relating the asymptotic trajectories of the algorithm of (4.9) and (4.15) to the solutions of a system of ordinary differential equations (ODE), when the gain matrix is of the form

$$\gamma(n) = \begin{bmatrix} 1/n & 0 \\ 0 & 1/n \end{bmatrix}. \quad (4.16)$$

This form of the gain matrix is restrictive since it corresponds to infinite memory for the adaptive algorithm, and therefore does not allow the tracking of time-varying parameters. But the application of Ljung's approach is nevertheless instructive since it relates formally the joint LMS algorithm to the joint steepest-descent algorithm.

A heuristic discussion about the method, based on the material presented in [53] and [4], is given in Appendix B. The method has been called the ODE approach and is used here to assess the convergence of the joint LMS algorithm. Define

$$f[\boldsymbol{\theta}_D(\tau)] = E[\boldsymbol{\psi}(n)\epsilon(n, d_n)] \quad (4.17)$$

and

$$G[\boldsymbol{\theta}_D(\tau)] = E[\boldsymbol{\psi}(n)\boldsymbol{\psi}^T(n)], \quad (4.18)$$

where $\boldsymbol{\theta}_D(\tau)$ is the mapping of $\boldsymbol{\theta}(n)$, using the following transformation

$$\tau = \sum_{k=1}^n 1/k, \quad (4.19)$$

and the expected value is taken over the input random variables. Then, the associated ODE is [4]

$$\begin{aligned}\frac{d\boldsymbol{\theta}_D(\tau)}{d\tau} &= \mathbf{R}^{-1}(\tau)f[\boldsymbol{\theta}_D(\tau)] \\ \frac{d\mathbf{R}(\tau)}{d\tau} &= G[\boldsymbol{\theta}_D(\tau)] - \mathbf{R}(\tau),\end{aligned}\tag{4.20}$$

The following theorems are given in [53] and proven in [52].

Theorem 4.1. *Let D_s denote the stability domain for $\boldsymbol{\theta}(n)$ such that the dynamical systems giving rise to $\boldsymbol{\psi}(n)$ and $\boldsymbol{\epsilon}(n)$ are stable. Subject to the boundedness conditions $\boldsymbol{\theta}(n) \in D_s$ and $|\boldsymbol{\psi}(n)| < C$ infinitely often a.s., where C is a random variable, and to the Lyapunov condition requiring the existence of a positive twice differentiable function V whose time derivative along the solutions of (4.20) satisfies*

$$\frac{dV}{d\tau} \leq 0, \quad \text{for } \boldsymbol{\theta}_D \in D_s, \quad \mathbf{R} > 0\tag{4.21}$$

then either (i)

$$\lim_{n \rightarrow \infty} \boldsymbol{\theta}(n) \in D_c \quad \text{w.p.1}\tag{4.22}$$

where

$$D_c = \left\{ \boldsymbol{\theta}_D, \mathbf{R} \mid \boldsymbol{\theta}_D \in D_s, \frac{dV}{d\tau} = 0 \right\},\tag{4.23}$$

or (ii) $\{\boldsymbol{\theta}(n)\}$ has a cluster point on the boundary of D_s . ■

Theorem 4.2. *The trajectories of the ODE (4.20) are the asymptotic paths of the estimates generated by the algorithm of (4.9) and (4.15). ■*

Consider (4.13) and (4.14) for a certain value $\boldsymbol{\theta}$. Then, from (4.17), and for a Type I system

$$\begin{aligned}f(\boldsymbol{\theta}) &= \begin{bmatrix} 2E[\mathbf{u}(nT - d_n)\boldsymbol{\epsilon}(n, d_n)] \\ 2E[\dot{\mathbf{y}}(n, d_n)\boldsymbol{\epsilon}(n, d_n)] \end{bmatrix} \\ &= \begin{bmatrix} -\nabla_{\mathbf{w}}\xi \\ -\frac{\partial \xi}{\partial d} \end{bmatrix} \\ &= -\nabla_{\boldsymbol{\theta}}\xi.\end{aligned}\tag{4.24}$$

The same result is obtained for the two other forms of (4.14). Considering the MSE function as a function of $\boldsymbol{\theta}_D(\tau)$, its derivative with respect to τ is

$$\begin{aligned}\frac{d\xi(\boldsymbol{\theta}_D(\tau))}{d\tau} &= \nabla_{\boldsymbol{\theta}}\xi(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_D(\tau)} \frac{d\boldsymbol{\theta}_D(\tau)}{d\tau} \\ &= -\mathbf{f}^T(\boldsymbol{\theta}_D(\tau))f(\boldsymbol{\theta}_D(\tau)) \\ &= -|f(\boldsymbol{\theta}_D(\tau))|^2,\end{aligned}\tag{4.25}$$

where (4.20) and (4.24) were used and $\mathbf{R}(n) = \mathbf{I}$.

Assuming that the observed signals are generated by stable dynamical systems and that the boundedness conditions of Theorem 4.1 are satisfied (if they are not satisfied, the algorithm is not of practical interest), then the function V can be taken to be the MSE function since its time derivative is given by (4.25) and is negative. Therefore, from Theorems 4.1 and 4.2, the vector $\boldsymbol{\theta}(n)$ converges to locally stable stationary points of the MSE function, since $f(\boldsymbol{\theta}_D(\tau))$ has to be of squared magnitude zero when $\boldsymbol{\theta}(n) \in D_c$ which, from (4.24), is true only when the necessary condition of (3.36) of Chapter 3 is respected.

Therefore, by using the ODE method, it is shown that when the adaptation factors μ_n and α_n both tend to zero, the joint LMS algorithm converges to a local minimum of the MSE function, like the exact version of the joint steepest-descent algorithm. This result, even if it does not apply directly to algorithm (4.8), is important by itself since it shows that if the adaptation factors are chosen sufficiently small, the estimates produced by the algorithm will be, on average, close to a stable stationary point of the MSE function. Furthermore, the above result shows that if the gain factors are constant but small, convergence cannot be attained in the sense that there exists an integer N such that $\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n)$ for $N \leq n$, but the difference between the parameter estimate and a stable stationary point will be small as n becomes large and can be made smaller by decreasing the gain factors.

Therefore, the ODE method, although applicable in a restrictive context, can justify, at least partially, the assumption of convergence of the joint LMS algorithm to solutions close to those of the joint steepest-descent algorithm.

4.3 Analysis of the Joint LMS Algorithm in Steady-State

The quality of the joint LMS algorithm can be studied by considering the quality of the two estimates that it generates. The delay and weight vector estimates being random variables, the joint algorithm can be analysed in terms of convergence in the mean and in the mean square of either estimate. Because of the coupling between the two adaptive processes, the gradient noise will affect the delay tracking and the derivative noise will itself influence the adaptive filter. These mutual effects can be included in the delay variance and weight noise vector correlation matrix, in steady-state conditions. The bounds for μ and α will be determined, for both types of convergence, and for the three forms of joint algorithms defined by (4.7) and (4.8). In every case, the analysis of the delay estimator is performed first. Then the weight vector estimator is considered and finally the two analyses are combined together, to obtain some misadjustment expressions for the joint LMS algorithm.

Such a separation of the analysis is artificial, but it allows the determination of tractable results.

In the course of the analyses, in addition to the general real signals and systems assumption mentioned in the introduction, the following assumptions are used:

1. The input and noise signals are zero-mean Gaussian random processes. The noise signals are also assumed to be white noise processes.
2. The adaptive system is in *steady-state* and the reference system is stationary, i.e. the reference delay is constant at $D_n = D$ and the reference filter is also fixed in time.
3. Independence theory holds, i.e. the zero-mean input data vectors are uncorrelated with each other and with $r(k)$. Then

$$\begin{aligned} E[\mathbf{u}(n)\mathbf{u}^T(k)] &= \mathbf{0} & \text{for } k = 0, 1, \dots, n-1 \\ E[\mathbf{u}(n)r(k)] &= \mathbf{0} & \text{for } k = 0, 1, \dots, n-1. \end{aligned} \quad (4.26)$$

The terminology *independence theory* is common in the analysis of adaptive algorithms (see [7] for example).

4. In steady-state, the adaptive weight vector \mathbf{w}_n can be expressed as

$$\mathbf{w}_n = \mathbf{w}_{\text{opt}} + \boldsymbol{\eta}_n \quad (4.27)$$

where \mathbf{w}_{opt} is the optimum Weiner solution given by

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{p}_n |_{d_n=D} \quad (4.28)$$

and $\boldsymbol{\eta}_n$ is a noise weight vector.

5. In the analysis of the delay estimator, the vector $\boldsymbol{\eta}_n$ is a zero-mean Gaussian vector, uncorrelated with the data vectors (because of (4.26)) and such that

$$E[\eta_i \eta_j] = 0 \quad \text{for } i \neq j. \quad (4.29)$$

The noise vector correlation matrix, defined as

$$\mathbf{K}_\eta = E[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T], \quad (4.30)$$

is therefore diagonal with the values $E[\eta_i^2(n)]$ on the main diagonal.

6. The system is in cancellation configuration. The results can be extended in a straightforward manner to the equalization case.
7. When the signal-to-noise ratios are assumed high, the adaptive filter Wiener solution is approximately equal to the reference filter (in practice, this amounts to SNR's greater than 10 dB).

Note that Assumption 3 can hardly be justified in practice, but has been used with success in the analysis of stochastic algorithms [7]. The noise vector properties stated in Assumption 5 are of the same kind and will prove to be useful in the analyses. Note in particular, that \mathbf{K}_η was found to be approximately equal to $\mu \xi_{\text{min}} \mathbf{I}$ in [6], for the LMS algorithm. The use of the central limit theorem supports the Gaussian assumption about $\boldsymbol{\eta}_n$. This assumption is also commonly used in the analysis of the LMS algorithm [22], [27].

4.3.1 The Joint LMS Algorithm in Type I Configuration

As mentioned in the introduction, it is assumed that $\mathbf{u}_n = \mathbf{u}(nT - d_n)$, i.e. that any adaptive delay modification is reflected on every sample of the adaptive filter delay line. This simplifies the analyses by making the input signal autocorrelation matrix time-invariant and by making the adaptive filter output equal to

$$y(n, d_n) = \mathbf{w}_n^T \mathbf{u}(nT - d_n). \quad (4.31)$$

The joint LMS algorithm is then of the form (for real signals)

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu e(n, d_n) \mathbf{u}(nT - d_n) \quad (4.32)$$

$$d_{n+1} = d_n + 2\alpha e(n, d_n) \mathbf{w}_n^T \dot{\mathbf{u}}(nT - d_n). \quad (4.33)$$

4.3.1.1 Analysis for the LMS Delay Estimator in Steady-State

The LMS delay tracking algorithm, in (4.33), is analyzed in terms of convergence of the delay estimate, in the mean and in the mean square. The following analysis parallels and extends that of Messer [29].

For $d_n = D$, the output of the adaptive branch can be expressed as

$$y(n, D) = \mathbf{w}_{\text{opt}}^T \mathbf{u}(nT - D) + \boldsymbol{\eta}_n^T \mathbf{u}(nT - D). \quad (4.34)$$

The first term on the right is defined as the optimum output $\hat{r}(n)$, since it represents the adaptive branch output for perfect modelling in the MSE sense. The second term on the right is defined as the output steady-state noise $\chi(n, D)$. Define $e_{\min}(n, D)$ as the error between the optimum adaptive branch and the reference branch, i.e.

$$e_{\min}(n, D) = r(n) - \hat{r}(n), \quad (4.35)$$

and the corresponding MSE as (also given in equation (3.53))

$$\xi_{\min} = E[e_{\min}^2(n, D)]. \quad (4.36)$$

Note that because of Assumption 5, the steady-state noise output is zero-mean and uncorrelated with $\hat{r}(n)$ and $r(n)$. In effect,

$$\begin{aligned} E[\hat{r}(n)\chi(n, D)] &= E[\mathbf{w}_{\text{opt}}^T \mathbf{u}(nT - D) \boldsymbol{\eta}_n^T \mathbf{u}(nT - D)] \\ &= \mathbf{w}_{\text{opt}}^T E[\mathbf{u}(nT - D) \mathbf{u}^T(nT - D)] E[\boldsymbol{\eta}_n] \\ &= 0 \end{aligned} \quad (4.37)$$

and,

$$\begin{aligned}
E[r(n)\chi(n, D)] &= E[(\hat{r}(n) + e_{\min}(n, D))\chi(n, D)] \\
&= E[e_{\min}(n, D)\boldsymbol{\eta}_n^T \mathbf{u}(nT - D)] \\
&= E[e_{\min}(n, D)\mathbf{u}^T(nT - D)]E[\boldsymbol{\eta}_n] \\
&= 0.
\end{aligned} \tag{4.38}$$

The approximation of equation (3.22) in Chapter 3 can be used with $\gamma_n(d_n) = e^2(n, d_n)$ and $\Theta_n = D$ (for real signals). Then

$$\begin{aligned}
\dot{\gamma}_n(D) &= 2e(n, D)\dot{e}(n, D) \\
&= -2e(n, D)\dot{y}(n, D)
\end{aligned} \tag{4.39}$$

and

$$\begin{aligned}
\ddot{\gamma}_n(D) &= 2\dot{e}^2(n, D) + 2e(n, D)\ddot{e}(n, D) \\
&= 2\dot{y}^2(n, D) - 2e(n, D)\ddot{y}(n, D).
\end{aligned} \tag{4.40}$$

In (4.39) and (4.40), the dot and double dot denote respectively the first and second derivative with respect to d_n . Therefore, using equation (3.22), the MSE estimate can be approximated as

$$\begin{aligned}
\gamma_n(d_n) &= e^2(n, d_n) \\
&\approx e^2(n, D) - 2(d_n - D)e(n, D)\dot{y}(n, D) + (d_n - D)^2[\dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D)].
\end{aligned} \tag{4.41}$$

The derivative estimate is then (for real signals)

$$\begin{aligned}
\hat{\nabla}_{d_n} \xi_n &= \frac{\partial \gamma_n(d_n)}{\partial d_n} = \frac{\partial e^2(n, d_n)}{\partial d_n} \\
&= -2e(n, D)\dot{y}(n, D) + 2(d_n - D)[\dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D)].
\end{aligned} \tag{4.42}$$

If the derivative noise N_n is defined as

$$N_n \triangleq \hat{\nabla}_{d_n} \xi_n|_{d_n=D}, \tag{4.43}$$

then, combining (4.42) and (4.43), the derivative noise is expressed as

$$N_n = -2e(n, D)\dot{y}(n, D) \tag{4.44}$$

and represents the error between $\hat{\nabla}_{d_n} \xi_n$ and $\partial \xi_n / \partial d_n$, when $d_n = D$. Defining the quantity G_n as

$$\begin{aligned}
G_n &= 1/2\dot{N}_n \\
&= \dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D),
\end{aligned} \tag{4.45}$$

equation (4.42) can be expressed as

$$\begin{aligned}
\hat{\nabla}_{d_n} \xi_n &= 2(d_n - D)G_n + N_n \\
&= (d_n - D)\dot{N}_n + N_n
\end{aligned} \tag{4.46}$$

and the LMS delay tracking algorithm is therefore approximately expressed as

$$d_{n+1} = d_n - 2\alpha(d_n - D)G_n - \alpha N_n. \tag{4.47}$$

Convergence in the Mean of the Delay Estimate

Take the expected value on both sides of (4.47), and rearrange. The result is

$$E[d_{n+1}] = E[(1 - 2\alpha G_n)d_n] + 2D\alpha E[G_n] - \alpha E[N_n]. \quad (4.48)$$

The following proposition simplifies expression (4.48).

Proposition 4.1. d_n is uncorrelated with the derivative noise N_n and its rate of change G_n . ■

Proof: From (4.47), it is seen that d_n is a function of d_{n-1} , G_{n-1} and N_{n-1} . But, from (4.44) and (4.45), G_{n-1} and N_{n-1} are functions of $r(n-1)$ and $y(n-1)$. The delay d_n is then a function of $r(n-1-i)$ and $y(n-1-i)$, for $i = 0, 1, \dots, n-1$. But G_n and N_n are functions of $r(n)$ and $y(n)$. From (4.32), the vector \mathbf{w}_n is a function of $\mathbf{u}(nT - T - d_{n-1}), \mathbf{u}(nT - 2T - d_{n-2}), \dots, \mathbf{u}(-d_0)$. In steady-state, $d_{n-i} \approx D$ and because of Assumption 3, \mathbf{w}_n is approximately uncorrelated with $\mathbf{u}(nT - d_n)$. This fact allows the following computation, for $k = 1, 2, \dots, n$,

$$\begin{aligned} E[y(n)y(n-k)] &= E[\mathbf{w}_n^T \mathbf{u}(nT - d_n) \mathbf{w}_{n-k}^T \mathbf{u}(nT - kT - d_{n-k})] \\ &= E[\mathbf{u}^T(nT - d_n)] E[\mathbf{w}_n^T \mathbf{w}_{n-k} \mathbf{u}(nT - kT - d_{n-k})] \\ &= 0, \end{aligned} \quad (4.49)$$

since $u(n)$ is zero-mean. If the signal portion of $r(n)$ is obtained by filtering $u(n)$ with an FIR filter of length equal to the adaptive filter length and since the noisy portion is white, then Assumption 3 implies that $E[r(n)r(n-k)] = E[r(n)y(n-k)] = 0$, for $k = 1, 2, \dots, n$. Therefore d_n is uncorrelated with G_n and N_n . ■

Equation (4.48) becomes

$$E[d_{n+1}] = (1 - 2\alpha E[G_n])E[d_n] + 2D\alpha E[G_n] - \alpha E[N_n]. \quad (4.50)$$

In Appendix D, $E[G_n]$ is found to be

$$E[G_n] = -\phi''_{rr}(0), \quad (4.51)$$

and

$$E[N_n] = 0 \quad (4.52)$$

because

$$\begin{aligned} E[N_n] &= \frac{\partial}{\partial d_n} E[e^2_{\min}(n, d_n)]|_{d_n=D} \\ &= \dot{\xi}_{\min} \\ &= 0. \end{aligned} \quad (4.53)$$

Then (4.50) simplifies to

$$E[d_{n+1}] = (1 + 2\alpha\phi''_{\hat{r}\hat{r}}(0))E[d_n] - 2D\alpha\phi''_{\hat{r}\hat{r}}(0). \quad (4.54)$$

Note that, for a Type I or a Type II-DAB adaptive system,

$$\begin{aligned} \ddot{\xi}_{\min} &= \frac{\partial^2 E[e_{\min}^2(n, d_n)]}{\partial d_n^2} \Big|_{d_n=D} \\ &= \frac{\partial^2 E[(r(n) - \mathbf{w}_{\text{opt}}^T \mathbf{u}(nT - d_n))^2]}{\partial d_n^2} \Big|_{d_n=D} \\ &= -2 \frac{\partial^2 E[r(n)\mathbf{w}_{\text{opt}}^T \mathbf{u}(nT - d_n)]}{\partial d_n^2} \Big|_{d_n=D} + \frac{\partial^2 (\mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}})}{\partial d_n^2} \\ &= -2 \frac{\partial^2 \phi_{r\hat{r}}(0)}{\partial d_n^2} + \frac{\partial^2 (\mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}})}{\partial d_n^2}. \end{aligned} \quad (4.55)$$

The second term on the right is zero and the quantity $\ddot{\xi}_{\min}$ is

$$\ddot{\xi}_{\min} = -2\ddot{\phi}_{r\hat{r}}(0). \quad (4.56)$$

This result is also valid for a Type II-DRB adaptive system. Using the results of Appendix C, (4.56) can also be expressed as

$$\begin{aligned} \ddot{\xi}_{\min} &= -2\phi''_{r\hat{r}}(0) \\ &\approx -2\phi''_{\hat{r}\hat{r}}(0) \end{aligned} \quad (4.57)$$

for high signal-to-noise ratios [†]. Note that because of orthogonality principles [7]

$$\begin{aligned} \phi_{r\hat{r}}(0) &= E[r(n)\hat{r}(n)] \\ &= E[(\hat{r}(n) + e_{\min}(n, D))\hat{r}(n)] \\ &= E[\hat{r}^2(n)] \\ &= \phi_{\hat{r}\hat{r}}(0). \end{aligned} \quad (4.58)$$

Therefore, (4.54) can be written as

$$E[d_{n+1}] = (1 - \alpha\ddot{\xi}_{\min})E[d_n] + D\alpha\ddot{\xi}_{\min}, \quad (4.59)$$

which shows the same form as the SD delay tracking algorithm of (3.25) with $\ddot{\gamma}_n(\Theta_n) = \ddot{\xi}_{\min}$ and $\Theta_n = D$.

Equation (4.59) converges if $|1 - \alpha\ddot{\xi}_{\min}| \leq 1$, and from the above derivations, the following proposition emerges.

[†] Note that $\phi''_{r\hat{r}}(0) = \phi''_{\hat{r}\hat{r}}(0)$ when the input and the noise signals are white.

Proposition 4.2. *In steady-state conditions, the delay estimator, given by the LMS delay tracking algorithm operating jointly with an adaptive filter in Type I configuration, is an unbiased estimator if*

$$0 < \alpha < \frac{2}{\xi_{\min}}. \quad (4.60)$$

■

Note that, in interpreting Proposition 4.2, it is important to keep in mind that the result is true if no false lock happens, i.e. if no noise samples force the delay estimate to lock on a local solution, or if the adaptive filter does not compensate at all for the delay reference. In this case, the first order linearized model leading to (4.47) applies and Proposition 4.2 can be used.

Convergence in the Mean Square of the Delay Estimate

Subtract the value D from each side of (4.47) and rearrange. This gives

$$d_{n+1} - D = (1 - 2\alpha G_n)(d_n - D) - \alpha N_n. \quad (4.61)$$

Square each side of (4.61) and take the expected value

$$E[(d_{n+1} - D)^2] = E[(1 - 2\alpha G_n)^2 (d_n - D)^2] - 2\alpha E[(1 - 2\alpha G_n)(d_n - D)N_n] + \alpha^2 E[N_n^2]. \quad (4.62)$$

Use Proposition 4.1, which states that d_n is uncorrelated with G_n and N_n . Equation (4.62) simplifies to

$$E[(d_{n+1} - D)^2] = E[(1 - 2\alpha G_n)^2] E[(d_n - D)^2] - 2\alpha E[(1 - 2\alpha G_n)N_n] E[(d_n - D)] + \alpha^2 E[N_n^2]. \quad (4.63)$$

It can be shown that $E[(1 - 2\alpha G_n)N_n] = 0$ (Appendix D) and, defining the time-varying delay estimate variance v_n as

$$v_n = E[(d_n - D)^2], \quad (4.64)$$

equation (4.63) simplifies to

$$v_{n+1} = E[(1 - 2\alpha G_n)^2] v_n + \alpha^2 E[N_n^2]. \quad (4.65)$$

Equation (4.65) indicates that there is convergence in the mean square sense if

$$|E[(1 - 2\alpha G_n)^2]| \leq 1. \quad (4.66)$$

Using the result of (4.51), the expected value is equal to

$$\begin{aligned} E[(1 - 2\alpha G_n)^2] &= 1 - 4\alpha E[G_n] + 4\alpha^2 E[G_n^2] \\ &= 1 - 2\alpha \xi_{\min} + 4\alpha^2 E[G_n^2]. \end{aligned} \quad (4.67)$$

The value of $E[G_n^2]$ is found to be (Appendix D)

$$\begin{aligned} E[G_n^2] &\approx 3(\phi_{\hat{r}\hat{r}}''(0))^2 + 6\phi_{\hat{r}\hat{r}}''(0)\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + 3(\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta])^2 \\ &\quad + (\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta]) \\ &\quad + 2\phi_{\hat{r}\hat{r}}''(0)(\phi_{rr}''(0) - \phi_{\hat{r}\hat{r}}''(0) - \phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta]), \end{aligned} \quad (4.68)$$

which, for high signal-to-noise ratios, can be approximated by (using (4.57))

$$\begin{aligned} E[G_n^2] &\approx 3(\phi_{\hat{r}\hat{r}}''(0))^2 + 4\phi_{\hat{r}\hat{r}}''(0)\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + 3(\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta])^2 \\ &\quad + (\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta]), \end{aligned} \quad (4.69)$$

where $\text{tr}[\cdot]$ is the trace operator, \mathbf{K}_η is the weight noise correlation matrix defined in (4.30) and $\phi^{(4)}(0)$ denotes $\partial^4\phi(\tau)/\partial\tau^4$ at $\tau = 0$. The condition of (4.66) can be expressed as

$$-1 < 1 - 2\alpha\ddot{\xi}_{\min} + 4\alpha^2 E[G_n^2] < 1 \quad (4.70)$$

or

$$0 < 1 - \alpha\ddot{\xi}_{\min} + 2\alpha^2 E[G_n^2] < 1. \quad (4.71)$$

Equation (4.71) is true if the following conditions are simultaneously verified

$$\begin{cases} 0 < 1 - \alpha\ddot{\xi}_{\min} + 2\alpha^2 E[G_n^2] & \text{(a)} \\ -1/2\ddot{\xi}_{\min} + \alpha E[G_n^2] < 0 & \text{(aa)}. \end{cases}$$

Condition (a) implies that a quadratic function in α must always be positive. This is true if the following condition is met

$$1 < \frac{8E[G_n^2]}{\ddot{\xi}_{\min}^2}. \quad (4.72)$$

Using (4.57) in (4.69), the numerator is given by

$$8E[G_n^2] \approx 6\ddot{\xi}_{\min}^2 + K,$$

where K is always positive. Therefore, (4.72) is always true and condition (a) is always verified for any value of α . This leaves condition (aa) to fix the range of α for convergence in the mean square. The following proposition has therefore been proven.

Proposition 4.3. *In steady-state conditions, the delay estimator, given by the LMS delay tracking algorithm operating jointly with an adaptive filter in Type I configuration, is convergent in the mean square if*

$$0 < \alpha < \frac{\ddot{\xi}_{\min}}{2E[G_n^2]}, \quad (4.73)$$

where the quantity $E[G_n^2]$ is given in (4.68). ■

Because, in steady-state, the expected values in (4.65) are time-invariant, the steady-state delay estimate variance is given by

$$\begin{aligned} v_{ss} &= \lim_{n \rightarrow \infty} v_n \\ &= \frac{\alpha^2 E[N_n^2]}{1 - E[(1 - 2\alpha G_n)^2]} \\ &= \frac{\alpha E[N_n^2]}{2\ddot{\xi}_{\min} - 4\alpha E[G_n^2]}, \end{aligned} \quad (4.74)$$

where $E[N_n^2]$ can be shown to be (Appendix D)

$$E[N_n^2] = -4(\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi''_{\hat{r}\hat{r}}(0) + \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta]). \quad (4.75)$$

Note that the steady-state variance is approached at the fastest rate when the quantity $E[(1 - 2\alpha G_n)^2]$ in (4.65) is minimum. This happens when the adaptation constant is

$$\alpha_{\text{opt}} = \frac{\ddot{\xi}_{\min}}{4E[G_n^2]}, \quad (4.76)$$

which is one half the maximum adaptation constant allowed by (4.73).

4.3.1.2 Analysis for the LMS Adaptive Filter in Steady-State

As with the LMS delay tracking algorithm, the LMS weight vector adaptive algorithm of (4.32) can be analyzed in terms of convergence in the mean and the mean square of the weight vector estimate.

Convergence in the Mean of the Weight Vector Estimate

Take the expected value of each side of the first equation of (4.32). The result is

$$\begin{aligned} E[\mathbf{w}_{n+1}] &= E[\mathbf{w}_n] + 2\mu E[e(n, d_n)\mathbf{u}(nT - d_n)] \\ &= E[\mathbf{w}_n] + 2\mu(E[r(n)\mathbf{u}(nT - d_n)] - E[\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_n]). \end{aligned} \quad (4.77)$$

From equation (3.7), the second expectation on the right hand side of (4.77) is equal to \mathbf{p}_n . But the cross-correlation vector is a function of the delay d_n , which is a random variable in the joint LMS algorithm. Therefore, \mathbf{p}_n is now a conditional expectation, conditioned on d_n and $E[r(n)\mathbf{u}(nT - d_n)]$ is equal to $E[\mathbf{p}_n]$, with the expectation taken with respect to the adaptive delay value.

From (4.32), it is noticed that the estimated weight vector \mathbf{w}_n is a function of the past input vectors $\mathbf{u}(nT - T - d_{n-1}), \mathbf{u}(nT - 2T - d_{n-2}), \dots, \mathbf{u}(-d_0)$. Assuming that independence

theory holds, since $d_{n-i} \approx D$ in steady-state, the weight vector \mathbf{w}_n is uncorrelated with $\mathbf{u}(nT - d_n)$ and the third expectation on the right hand side of (4.77) can be expressed as

$$\begin{aligned} E[\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_n] &= E[\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)]E[\mathbf{w}_n] \\ &= \mathbf{R}E[\mathbf{w}_n]. \end{aligned} \quad (4.78)$$

Therefore, (4.77) can be expressed as

$$E[\mathbf{w}_{n+1}] = E[\mathbf{w}_n] + 2\mu(E[\mathbf{p}_n] - \mathbf{R}E[\mathbf{w}_n]). \quad (4.79)$$

In order to compute $E[\mathbf{p}_n]$, express \mathbf{p}_n as a function of d_n , as in equation (3.54). This expression is

$$\phi_{ur}(-jT - d_n) \approx \phi_{ur}(-jT - D_n) + (d_n - D_n)\dot{\phi}_{ur}(-jT - D_n) + \frac{(d_n - D_n)^2}{2}\ddot{\phi}_{ur}(-jT - D_n). \quad (4.80)$$

Therefore, because the delay estimator is assumed unbiased and in steady-state, the expected value of the cross-correlation vector is

$$\begin{aligned} E[\mathbf{p}_n] &\approx \mathbf{p}(D) + 1/2v_{ss}\ddot{\mathbf{p}}(D) \\ &= \mathbf{R}\mathbf{w}_{\text{opt}} + 1/2v_{ss}\ddot{\mathbf{p}}(D), \end{aligned} \quad (4.81)$$

where v_{ss} is the steady-state delay estimate variance, and equation (4.28) was used. Using (4.81) in (4.79) gives

$$E[\mathbf{w}_{n+1}] = (\mathbf{I} - 2\mu\mathbf{R})E[\mathbf{w}_n] + \mu(2\mathbf{R}\mathbf{w}_{\text{opt}} + v_{ss}\ddot{\mathbf{p}}(D)). \quad (4.82)$$

This equation can also be expressed as

$$E[\mathbf{w}_{n+1}] = (\mathbf{I} - 2\mu\mathbf{R})^n E[\mathbf{w}_0] + \mathbf{w}_{\text{opt}} + 1/2v_{ss}\mathbf{R}^{-1}\ddot{\mathbf{p}}(D)[\mathbf{I} - (\mathbf{I} - 2\mu\mathbf{R})^n]. \quad (4.83)$$

Therefore, $E[\mathbf{w}_n]$ converges to $\mathbf{w}_{\text{opt}} + 1/2v_{ss}\mathbf{R}^{-1}\ddot{\mathbf{p}}(D)$ if the gain factor μ is smaller than the inverse of the maximum eigenvalue of \mathbf{R} [7]. This can be formalized in the following proposition.

Proposition 4.4. *In steady-state conditions, the weight vector estimator, given by the adaptive filter LMS algorithm operating jointly with a mean square convergent delay tracking algorithm in Type I configuration, converges in the mean if*

$$0 < \mu < \frac{1}{\lambda_{\max}}, \quad (4.84)$$

where λ_{\max} denotes the maximum value of the input signal autocorrelation matrix \mathbf{R} . The weight vector estimate experiences a bias given by

$$\mathbf{b} = 1/2v_{ss}\mathbf{R}^{-1}\ddot{\mathbf{p}}(D). \quad (4.85)$$

■

Note that the convergence condition of (4.84) is identical to the usual condition for convergence in the mean of an LMS adaptive filter [7].

Convergence in the Mean Square of the Weight Vector Estimate

The weight noise vector correlation matrix $\mathbf{K}_\eta(n+1)$, at iteration $n+1$, is computed in this section and a condition for its convergence, in the matrix norm sense, to a finite steady-state value is established. From equations (4.27) and (4.32), the noise vector can be written as

$$\begin{aligned}
 \boldsymbol{\eta}_{n+1} &= \mathbf{w}_{n+1} - \mathbf{w}_{\text{opt}} \\
 &= \mathbf{w}_n + 2\mu e(n, d_n) \mathbf{u}(nT - d_n) - \mathbf{w}_{\text{opt}} \\
 &= [\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)] \boldsymbol{\eta}_n \\
 &\quad + 2\mu [\mathbf{u}(nT - d_n) r(n) - \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \mathbf{w}_{\text{opt}}].
 \end{aligned} \tag{4.86}$$

Then, $\mathbf{K}_\eta(n+1)$ can be expressed as

$$\begin{aligned}
 \mathbf{K}_\eta(n+1) &= E[\boldsymbol{\eta}_{n+1} \boldsymbol{\eta}_{n+1}^T] \\
 &= E[(\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)) \boldsymbol{\eta}_n \boldsymbol{\eta}_n^T \\
 &\quad (\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n))^T] \\
 &\quad + 2\mu E[(\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)) \boldsymbol{\eta}_n \\
 &\quad (\mathbf{u}(nT - d_n) r(n) - \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \mathbf{w}_{\text{opt}})^T] \\
 &\quad + 2\mu E[(\mathbf{u}(nT - d_n) r(n) - \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \mathbf{w}_{\text{opt}}) \\
 &\quad \boldsymbol{\eta}_n^T (\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n))^T] \\
 &\quad + 4\mu^2 E[(\mathbf{u}(nT - d_n) r(n) - \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \mathbf{w}_{\text{opt}}) \\
 &\quad (\mathbf{u}(nT - d_n) r(n) - \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \mathbf{w}_{\text{opt}})^T].
 \end{aligned} \tag{4.87}$$

The four terms of equation (4.87) can be evaluated individually as follows:

1st term.

$$\begin{aligned}
 &E[(\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)) \boldsymbol{\eta}_n \boldsymbol{\eta}_n^T (\mathbf{I} - 2\mu \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n))^T] \\
 &= E[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] \\
 &\quad - 2\mu E[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] E[\mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)] \\
 &\quad - 2\mu E[\mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)] E[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] \\
 &\quad + 4\mu^2 E[\mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \boldsymbol{\eta}_n \boldsymbol{\eta}_n^T \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)] \\
 &= \mathbf{K}_\eta(n) - 2\mu \mathbf{K}_\eta(n) \mathbf{R} - 2\mu \mathbf{R} \mathbf{K}_\eta(n) \\
 &\quad + 4\mu^2 E[\mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n) \boldsymbol{\eta}_n \boldsymbol{\eta}_n^T \mathbf{u}(nT - d_n) \mathbf{u}^T(nT - d_n)] \\
 &\approx \mathbf{K}_\eta(n) - 2\mu \mathbf{K}_\eta(n) \mathbf{R} - 2\mu \mathbf{R} \mathbf{K}_\eta(n) + 4\mu^2 \text{Rtr}[\mathbf{R} \mathbf{K}_\eta(n)],
 \end{aligned} \tag{4.88}$$

where the last step follows from the Gaussian and independence assumptions and can be carried out in details as in [54] (see also [7], pp. 221–224).

2nd term.

$$\begin{aligned}
& E\{(\mathbf{I} - 2\mu\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n))\boldsymbol{\eta}_n(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T\} \\
&= E\{\boldsymbol{\eta}_n(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T\} \\
&\quad - 2\mu E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\boldsymbol{\eta}_n(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T\} \\
&= E\{\boldsymbol{\eta}_n\}E\{(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T\} \\
&\quad - 2\mu E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\boldsymbol{\eta}_n\mathbf{u}^T(nT - d_n)r(n)\} \\
&\quad + 2\mu E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\boldsymbol{\eta}_n\mathbf{w}_{\text{opt}}^T\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\} \\
&\approx \mathbf{b}\mathbf{b}^T\mathbf{R} - 2\mu\mathbf{R}\mathbf{b}\mathbf{w}_{\text{opt}}^T\mathbf{R} - 2\mu\mathbf{R}\mathbf{b}\mathbf{b}^T\mathbf{R} + 2\mu\mathbf{R}\mathbf{b}\mathbf{w}_{\text{opt}}^T\mathbf{R} \\
&= \mathbf{b}\mathbf{b}^T\mathbf{R} - 2\mu\mathbf{R}\mathbf{b}\mathbf{b}^T\mathbf{R},
\end{aligned} \tag{4.89}$$

where (4.81) and (4.85) were used. But note that the vector \mathbf{b} is proportional to the delay estimate variance v_{ss} . Assuming that this variance is small, then the second term is approximately zero since it is proportional to the square of the variance.

3rd term.

The third term of equation (4.87) is the transpose of the second term and is therefore approximately zero.

4th term.

$$\begin{aligned}
& 4\mu^2 E\{(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}) \\
&\quad (\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T\} \\
&= 4\mu^2 E\{\mathbf{u}(nT - d_n)r(n)r(n)\mathbf{u}^T(nT - d_n)\} \\
&\quad - 4\mu^2 E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}\mathbf{u}^T(nT - d_n)r(n)\} \\
&\quad - 4\mu^2 E\{\mathbf{u}(nT - d_n)r(n)\mathbf{w}_{\text{opt}}^T\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\} \\
&\quad + 4\mu^2 E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}\mathbf{w}_{\text{opt}}^T\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\}.
\end{aligned} \tag{4.90}$$

Reasoning as in [7], the four expectations of equation (4.90) are found to be

$$E\{\mathbf{u}(nT - d_n)r(n)r(n)\mathbf{u}^T(nT - d_n)\} \approx \mathbf{R}\phi_{rr}(0) \tag{4.91}$$

$$E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}\mathbf{u}^T(nT - d_n)r(n)\} \approx \mathbf{R}E\{\mathbf{p}_n^T\}\mathbf{w}_{\text{opt}} \tag{4.92}$$

$$E\{\mathbf{u}(nT - d_n)r(n)\mathbf{w}_{\text{opt}}^T\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\} \approx \mathbf{R}\mathbf{w}_{\text{opt}}^T E\{\mathbf{p}_n\} \tag{4.93}$$

$$E\{\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}\mathbf{w}_{\text{opt}}^T\mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\} \approx \mathbf{R}\mathbf{w}_{\text{opt}}^T\mathbf{R}\mathbf{w}_{\text{opt}}. \tag{4.94}$$

Using (4.81), the fourth term is given by

$$\begin{aligned}
& 4\mu^2 E[(\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}}) \\
& \quad (\mathbf{u}(nT - d_n)r(n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{\text{opt}})^T] \\
& = 4\mu^2 \mathbf{R}[\phi_{rr}(0) - \mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}} - 1/2 v_{ss}(\ddot{\mathbf{p}}^T(D)\mathbf{w}_{\text{opt}} + \mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D))] \\
& = 4\mu^2 \mathbf{R}[\xi_{\min} + \ddot{\xi}_{\min} v_{ss}/2],
\end{aligned} \tag{4.95}$$

where ξ_{\min} is the minimum MSE attainable as defined in (4.36), and its second derivative with respect to the delay, when $d_n = D$, is defined in (4.57) and can take the form

$$\ddot{\xi}_{\min} = -[\ddot{\mathbf{p}}^T(D)\mathbf{w}_{\text{opt}} + \mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D)], \tag{4.96}$$

because

$$\begin{aligned}
\mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D) &= E[\mathbf{w}_{\text{opt}}^T \ddot{\mathbf{u}}(nT - D)r(n)] \\
&= E[(\partial^2 \hat{r} / \partial d_n^2)r(n)] \\
&= \phi''_{rr}(0) \\
&= -\ddot{\xi}_{\min}/2.
\end{aligned} \tag{4.97}$$

Collecting the four terms, the time evolution of the weight-error correlation matrix is

$$\mathbf{K}_\eta(n+1) = \mathbf{K}_\eta(n) - 2\mu[\mathbf{K}_\eta(n)\mathbf{R} + \mathbf{R}\mathbf{K}_\eta(n)] + 4\mu^2 \mathbf{R} \text{tr}[\mathbf{R}\mathbf{K}_\eta(n)] + 4\mu^2 \mathbf{R}[\xi_{\min} + \ddot{\xi}_{\min} v_{ss}/2]. \tag{4.98}$$

Except for the term involving the delay estimate variance, equation (4.98) is identical to the one for an adaptive filter operating alone ([7], equation (5.74)). In order to have convergence in the mean square of the weight vector estimate, the correlation matrix must stay bounded in some sense. The norm of this matrix can be used with that effect.

The norm of a matrix \mathbf{A} , denoted by $\|\mathbf{A}\|$, is the number defined by [55]

$$\|\mathbf{A}\| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}. \tag{4.99}$$

It can also be shown that $\|\mathbf{A}\|^2$ is equal to the largest eigenvalue of the product $\mathbf{A}^H \mathbf{A}$ [55]. When \mathbf{A} is an autocorrelation matrix, the norm $\|\mathbf{A}\|$ is then equal to the largest eigenvalue of \mathbf{A} . Note that the definition based on the largest eigenvalue is not necessarily unique.

Proceeding as in [7], equation (4.98) is first expressed in normal form by using the unitary similarity transformation

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T, \tag{4.100}$$

where \mathbf{Q} is a unitary matrix with the orthonormal eigenvectors of \mathbf{R} as columns and $\mathbf{\Lambda}$ is a diagonal matrix with the corresponding eigenvalues on the main diagonal. Using this transformation in (4.98), with

$$\mathbf{X}(n) = \mathbf{Q}^T \mathbf{K}_\eta(n) \mathbf{Q}, \tag{4.101}$$

gives

$$\mathbf{X}(n+1) = \mathbf{X}(n) - 2\mu[\mathbf{X}(n)\mathbf{\Lambda} + \mathbf{\Lambda}\mathbf{X}(n)] + 4\mu^2\mathbf{\Lambda}\text{tr}[\mathbf{\Lambda}\mathbf{X}(n)] + 4\mu^2\mathbf{\Lambda}[\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2]. \quad (4.102)$$

Because the matrix \mathbf{Q} is unitary, the norm of $\mathbf{K}_\eta(n)$ is equal to the norm of $\mathbf{X}(n)$. Therefore, the weight vector estimator converges in the mean square if and only if the largest eigenvalue of the matrix $\mathbf{X}(n)$, when n tends to infinity, is finite. Since an autocorrelation matrix is always nonnegative definite [7], the largest eigenvalue of $\mathbf{X}(n)$ is finite if and only if the trace of $\mathbf{K}_\eta(n)$, which is equal to the trace of $\mathbf{X}(n)$, is finite. A recursive equation for the diagonal element of $\mathbf{X}(n)$ can be obtained by proceeding as in [7], pp. 229–230. The relation is

$$\mathbf{x}(n) = \mathbf{B}^n[\mathbf{x}(0) - 4\mu^2(\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2)(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda}] + 4\mu^2(\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2)(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda}, \quad (4.103)$$

where the $M \times 1$ vectors $\mathbf{x}(n)$ and $\boldsymbol{\lambda}$ are defined as

$$\mathbf{x}(n) = [x_1(n), x_2(n), \dots, x_M(n)]^T \quad (4.104)$$

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_M]^T, \quad (4.105)$$

with the $x_i(n)$'s being the diagonal elements of the matrix $\mathbf{X}(n)$, the λ_i 's being the eigenvalues of the input signal autocorrelation matrix \mathbf{R} and the $M \times M$ matrix \mathbf{B} has elements defined as

$$b_{ij} = \begin{cases} (1 - 2\mu\lambda_i)^2 & i = j \\ 4\mu^2\lambda_i\lambda_j & i \neq j. \end{cases} \quad (4.106)$$

Since the matrix \mathbf{B} is symmetric, a unitary similarity transformation similar to that described in (4.100) can be found such that

$$\mathbf{B} = \mathbf{G}\mathbf{C}\mathbf{G}^T,$$

where the matrix \mathbf{C} is diagonal with elements that are the eigenvalues of \mathbf{B} . Therefore, (4.103) converges to its steady-state component

$$\mathbf{x}_{ss} = 4\mu^2(\xi_{\min} + \ddot{\xi}_{\min}v_n/2)(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\lambda} \quad (4.107)$$

if and only if the eigenvalues of matrix \mathbf{B} all have magnitude less than one. It is demonstrated in [7] that this is the case if and only if the parameter μ satisfies the condition

$$0 < \mu < \frac{1}{\text{tr}[\mathbf{R}]}.$$

Therefore, if the delay estimate variance v_{ss} is finite, the trace of the weight-error correlation matrix \mathbf{K}_η is finite and the condition for convergence in the mean square is given in the following proposition.

Proposition 4.5. *In steady-state conditions, the weight vector estimator, given by the adaptive filter LMS algorithm operating jointly with a mean square convergent delay tracking algorithm in Type I configuration, is convergent in the mean square if*

$$0 < \mu < \frac{1}{\sum_{i=1}^M \lambda_i}, \quad (4.108)$$

where λ_i is the i^{th} eigenvalue of the $M \times M$ input signal autocorrelation matrix \mathbf{R} . ■

This condition for convergence in the mean square sense is identical to the one for an adaptive filter operating alone.

From the similarity transformation of (4.101), with the matrix \mathbf{Q} being unitary, the trace of the matrix $\mathbf{K}_\eta(n)$ is equal to the trace of the matrix $\mathbf{X}(n)$. Then, from the definitions of the vector $\mathbf{x}(n)$ in (4.104), the following is true

$$\begin{aligned} \text{tr}[\mathbf{K}_\eta] &= \text{tr}[\mathbf{X}_{\text{ss}}] \\ &= \sum_{i=1}^M x_{i(\text{ss})}, \end{aligned} \quad (4.109)$$

where \mathbf{X}_{ss} is the steady-state version of $\mathbf{X}(n)$ and $x_{i(\text{ss})}$ is the i^{th} element of the corresponding steady-state version of $\mathbf{x}(n)$. The elements of the vector \mathbf{x}_{ss} can be found from (4.107), or by letting n tend to infinity in (4.103). The result is that every component of \mathbf{x}_{ss} is equal to [7]

$$x_{i(\text{ss})} = \frac{\mu(\xi_{\min} + \ddot{\xi}_{\min} v_{\text{ss}}/2)}{1 - \mu \text{tr}[\mathbf{R}]}. \quad (4.110)$$

Therefore,

$$\text{tr}[\mathbf{K}_\eta] = \mu M \frac{\xi_{\min} + \ddot{\xi}_{\min} v_{\text{ss}}/2}{1 - \mu \text{tr}[\mathbf{R}]}. \quad (4.111)$$

If the adaptation constant μ is small enough to make

$$\mu \text{tr}[\mathbf{R}] \ll 1,$$

then (4.111) can be written as

$$\begin{aligned} \text{tr}[\mathbf{K}_\eta] &\approx \mu M (\xi_{\min} + \ddot{\xi}_{\min} v_{\text{ss}}/2) \\ &= \text{tr}[\mathbf{K}'_\eta] + \mu M \ddot{\xi}_{\min} v_{\text{ss}}/2, \end{aligned} \quad (4.112)$$

where $\text{tr}[\mathbf{K}'_\eta]$ is defined as the trace of the weight-error correlation matrix specific to the adaptive filter and is given as

$$\text{tr}[\mathbf{K}'_\eta] = \mu M \xi_{\min}. \quad (4.113)$$

4.3.1.3 Excess Mean-Squared Error and Misadjustment with the Joint LMS Algorithm

The steady-state MSE, for the joint LMS algorithm, is first computed and the excess MSE is deduced. Then, a misadjustment expression is derived. From equation (3.5), the steady-state MSE function is

$$\xi_{ss} = \phi_{rr}(0) + E[\mathbf{w}_n^T \mathbf{R} \mathbf{w}_n] - 2E[\mathbf{w}_n^T \mathbf{p}_n]. \quad (4.114)$$

Equation (4.27) transforms (4.114) into

$$\xi_{ss} = \phi_{rr}(0) + \mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}} + E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] + 2E[\boldsymbol{\eta}_n^T \mathbf{R} \mathbf{w}_{\text{opt}}] - 2\mathbf{w}_{\text{opt}}^T E[\mathbf{p}_n] - 2E[\boldsymbol{\eta}_n^T \mathbf{p}_n]. \quad (4.115)$$

The use of (4.81) gives

$$\begin{aligned} \xi_{ss} &= \phi_{rr}(0) + \mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}} + E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] - 2\mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}} - v_{ss} \mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D) - \mathbf{b} v_{ss} \ddot{\mathbf{p}}(D) \\ &\approx \phi_{rr}(0) - \mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}} - v_{ss} \mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D) + E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] \\ &= \xi_{\min} + v_{ss} \ddot{\xi}_{\min} / 2 + E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n], \end{aligned} \quad (4.116)$$

where the expression $\phi_{rr}(0) - \mathbf{w}_{\text{opt}}^T \mathbf{R} \mathbf{w}_{\text{opt}}$ is explicitly defined as ξ_{\min} , the expression $\mathbf{w}_{\text{opt}}^T \ddot{\mathbf{p}}(D)$ is replaced by its equivalent given in (4.97) and the steady-state delay variance v_{ss} is assumed small. The last term of (4.116) can be expressed as

$$\begin{aligned} E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] &= \text{tr}[\mathbf{R} \mathbf{K}_{\boldsymbol{\eta}}] \\ &= \text{tr}[\boldsymbol{\Lambda} \mathbf{X}_{ss}] \\ &= \boldsymbol{\lambda}^T \mathbf{x}_{ss}. \end{aligned} \quad (4.117)$$

Combining (4.107) and (4.117) gives

$$E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] = 4\mu^2 (\xi_{\min} + \ddot{\xi}_{\min} v_{ss} / 2) \boldsymbol{\lambda}^T (\mathbf{I} - \mathbf{B})^{-1} \boldsymbol{\lambda}, \quad (4.118)$$

which can be shown to be equal to [7]

$$E[\boldsymbol{\eta}_n^T \mathbf{R} \boldsymbol{\eta}_n] = \frac{\mu (\xi_{\min} + \ddot{\xi}_{\min} v_{ss} / 2) \text{tr}[\mathbf{R}]}{1 - \mu \text{tr}[\mathbf{R}]}. \quad (4.119)$$

Use (4.119) into (4.116) gives the final expression for the joint MSE function

$$\xi_{ss} = \xi_{\min} + v_{ss} \ddot{\xi}_{\min} / 2 + \frac{\mu (\xi_{\min} + \ddot{\xi}_{\min} v_{ss} / 2) \text{tr}[\mathbf{R}]}{1 - \mu \text{tr}[\mathbf{R}]}. \quad (4.120)$$

The excess MSE is then

$$\begin{aligned} \xi_{\text{ex}} &= \xi_{ss} - \xi_{\min} \\ &= v_{ss} \ddot{\xi}_{\min} / 2 + \frac{\mu (\xi_{\min} + \ddot{\xi}_{\min} v_{ss} / 2) \text{tr}[\mathbf{R}]}{1 - \mu \text{tr}[\mathbf{R}]} \\ &= \xi_{\text{ex}}^{\text{d}} + \xi_{\text{ex}}^{\text{f}} + \xi_{\text{ex}}^{\text{df}}, \end{aligned} \quad (4.121)$$

where the excess MSE specific to the adaptive delay element is defined as

$$\xi_{\text{ex}}^{\text{d}} = \frac{v_{\text{ss}}\ddot{\xi}_{\text{min}}}{2}, \quad (4.122)$$

the excess MSE specific to the adaptive filter is defined as

$$\xi_{\text{ex}}^{\text{f}} = \frac{\mu\xi_{\text{min}}\text{tr}[\mathbf{R}]}{1 - \mu\text{tr}[\mathbf{R}]}, \quad (4.123)$$

and the cross-product excess MSE is defined as

$$\xi_{\text{ex}}^{\text{df}} = \frac{\mu\ddot{\xi}_{\text{min}}v_{\text{ss}}\text{tr}[\mathbf{R}]}{2(1 - \mu\text{tr}[\mathbf{R}])}. \quad (4.124)$$

Note that the expression for $\xi_{\text{ex}}^{\text{d}}$ is equal to the expected value of the excess MSE given in (3.61). This expression is also valid for pure LMS delay estimation [29] and the expression for $\xi_{\text{ex}}^{\text{f}}$ is valid for an adaptive LMS filter operating without an adaptive delay [7].

The misadjustment is defined as the ratio of the excess MSE to ξ_{min} . Therefore, the misadjustment expression is

$$\begin{aligned} \mathcal{M} &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{df}} \\ &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{d}}\mathcal{M}^{\text{f}} \end{aligned} \quad (4.125)$$

where the misadjustment specific to the adaptive delay element is defined as

$$\mathcal{M}^{\text{d}} = \frac{v_{\text{ss}}\ddot{\xi}_{\text{min}}}{2\xi_{\text{min}}}, \quad (4.126)$$

the misadjustment specific to the adaptive filter is defined as

$$\mathcal{M}^{\text{f}} = \frac{\mu\text{tr}[\mathbf{R}]}{1 - \mu\text{tr}[\mathbf{R}]}, \quad (4.127)$$

and the cross-product misadjustment is defined as

$$\begin{aligned} \mathcal{M}^{\text{df}} &= \frac{\mu\ddot{\xi}_{\text{min}}v_{\text{ss}}\text{tr}[\mathbf{R}]}{2\xi_{\text{min}}(1 - \mu\text{tr}[\mathbf{R}])} \\ &= \mathcal{M}^{\text{d}}\mathcal{M}^{\text{f}}. \end{aligned} \quad (4.128)$$

4.3.2 The Joint LMS Algorithm in Type II Configuration: Delay in Adaptive Branch

The particularity of the Type II-DAB configuration is that the adaptation is a function of the delayed adaptive filter (see equation (3.10) and equations (3.33) to (3.35) in Chapter 3). The adaptive branch output is given in (4.4) and the corresponding joint LMS algorithm is

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n + 2\mu e(n, d_n)\mathbf{u}(nT - d_n) \\ d_{n+1} &= d_n + 2\alpha e(n, d_n)\dot{y}(nT - d_n), \end{aligned} \quad (4.129)$$

where $e(n, d_n)$ is given in (4.3).

4.3.2.1 Analysis for the LMS Delay Estimator in Steady-State

Because the output of the adaptive branch is given by (4.4), the output steady-state noise $\chi(n, D)$ is given by

$$\chi(n, D) = \boldsymbol{\eta}_{nT-D}^T \mathbf{u}(nT - D), \quad (4.130)$$

and the derivative of $\chi(n, D)$ with respect to the delay is a function of the derivatives of both $\boldsymbol{\eta}_{nT-D}^T$ and $\mathbf{u}(nT - D)$. This fact does not affect, for the most part, the derivations of the convergence conditions presented in Subsection 4.3.1.1. Proposition 4.2 is unchanged and Proposition 4.3 still holds with $E[G_n^2]$ given by

$$\begin{aligned} E[G_n^2] &\approx 3(\phi_{\hat{r}\hat{r}}''(0))^2 + 6\phi_{\hat{r}\hat{r}}''(0)[\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + \phi_{uu}(0) \sum_i \phi_{\eta_i\eta_i}''(0)] \\ &\quad + 3[\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + \phi_{uu}(0) \sum_i \phi_{\eta_i\eta_i}''(0)]^2 \\ &\quad + 2\phi_{\hat{r}\hat{r}}''(0)(\phi_{\hat{r}\hat{r}}''(0) - \phi_{\hat{r}\hat{r}}''(0) - [\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + \phi_{uu}(0) \sum_i \phi_{\eta_i\eta_i}''(0)]) \\ &\quad + (\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta]) \\ &\quad (\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta] + \phi_{uu}(0) \sum_i \phi_{\eta_i\eta_i}^{(4)}(0) + 6\phi_{uu}''(0) \sum_i \phi_{\eta_i\eta_i}''(0)). \end{aligned} \quad (4.131)$$

The steady-state delay variance is still given by (4.74) with

$$E[N_n^2] = -4(\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}''(0) + \phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta] + \phi_{uu}(0) \sum_i \phi_{\eta_i\eta_i}''(0)). \quad (4.132)$$

Note that equations (4.131) and (4.132) reduce to (4.68) and (4.75) when the adaptive weight vector is not a function of the delay ($\phi_{\eta_i\eta_i}''(0) = \phi_{\eta_i\eta_i}^{(4)}(0) = 0$).

The second derivative $\phi_{\eta_i\eta_i}''(0)$ can be approximated by Stirling's formula

$$\phi_{\eta_i\eta_i}''(0) \approx \frac{2\phi_{\eta_i\eta_i}(k) - 2\phi_{\eta_i\eta_i}(0)}{k^2}. \quad (4.133)$$

It is shown below that $\phi_{\eta_i\eta_i}(1) \approx \phi_{\eta_i\eta_i}(0)$, which, when used in (4.133), implies that $\phi_{\eta_i\eta_i}''(0) \approx 0$. This result can be heuristically explained by noting that if μ is small (as it is in practice), the correction made to the weight vector is small (see equation 4.129), and the autocorrelation of the noise vector components is approximately constant around a lag of zero. Therefore, the results of Subsection 4.3.1.1 can be used without any modifications, unless the adaptation factor μ is such that the approximation (obtained from (4.172))

$$1 + 2\mu\lambda_i \approx 1 \quad \forall i \quad (4.134)$$

is not true.

4.3.2.2 Analysis for the LMS Adaptive Filter in Steady-State

Some complications appear in the analysis of the LMS adaptive filter. From (4.129), the weight vector adaptation is performed according to

$$\begin{aligned}\mathbf{w}_{n+1} &= \mathbf{w}_n + 2\mu\epsilon(n, d_n)\mathbf{u}(nT - d_n) \\ &= \mathbf{w}_n + 2\mu[r(n)\mathbf{u}(nT - d_n) - \mathbf{u}(nT - d_n)\mathbf{u}^T(nT - d_n)\mathbf{w}_{nT-d_n}].\end{aligned}\quad (4.135)$$

This type of algorithm has been analysed for a *constant integer* delay [49], [56]. The use of a fractional and stochastic delay complicates greatly the problem. In order to simplify the analysis, it will be assumed, throughout Subsection 4.3.2.2, that the reference delay $D_* = D/T$ is an *integer*.

Convergence in the Mean of the Weight Vector Estimate when D_* is an Integer

Taking the expected value on each side of (4.135), making use of the independence assumption and using (4.81), the following equation is obtained for the update of the average weight vector

$$E[\mathbf{w}_{n+1}] = E[\mathbf{w}_n] - 2\mu\mathbf{R}\{E[\mathbf{w}_{nT-d_n}] - \mathbf{w}_{\text{opt}}\} + \mu v_{\text{ss}}\ddot{\mathbf{p}}(D). \quad (4.136)$$

Use the similarity transformation of (4.100) and define the normalized error vector \mathbf{E}_n and the normalized cross-correlation vector $\mathbf{c}(d_n)$ as

$$\begin{aligned}\mathbf{E}_n &= \mathbf{Q}^T \{E[\mathbf{w}_n] - \mathbf{w}_{\text{opt}}\} \\ \mathbf{c}(d_n) &= \mathbf{Q}^T \mathbf{p}(d_n).\end{aligned}\quad (4.137)$$

Equation (4.136) then becomes

$$E[\mathbf{E}_{n+1}] = E[\mathbf{E}_n] - 2\mu\Lambda E[\mathbf{E}_{nT-d_n}] + \mu v_{\text{ss}}\ddot{\mathbf{c}}(D). \quad (4.138)$$

Note that the expected values are taken over the input data, which amounts to expectations taken jointly over the adaptive weight vector and the adaptive delay. Denote an expected value with respect to the weight vector as $E_{\mathbf{w}}[\cdot]$ and an expectation taken with respect to the delay as $E_d[\cdot]$. Consider $\mathbf{w}_n(d_n)$ as a function of d_n . If the delay steady-state variance is small, $\mathbf{w}_n(d_n)$ can be represented approximately as

$$\mathbf{w}_n(d_n) \approx \mathbf{w}_n(D) + (d_n - D)\dot{\mathbf{w}}_n(D). \quad (4.139)$$

Since the delay estimate is unbiased, the expected value of $\mathbf{w}_n(d_n)$ with respect to d_n is [†]

$$E_d[\mathbf{w}_n(d_n)] \approx \mathbf{w}_n(D), \quad (4.140)$$

[†] Note that in order to be consistent with previous results, a term proportional to the delay variance v_{ss} should be present. But in a first analysis, this term is neglected in order to avoid expressions containing derivatives of the weight vector

and

$$E_d[\mathbf{E}_{nT-d_n}] \approx \mathbf{E}_{nT-D}. \quad (4.141)$$

Equation (4.138) then becomes

$$E_{\mathbf{w}}[\mathbf{E}_{n+1}] = E_{\mathbf{w}}[\mathbf{E}_n] - 2\mu\Lambda E_{\mathbf{w}}[\mathbf{E}_{nT-D}] + \mu v_{\text{ss}}\ddot{\mathbf{c}}(D). \quad (4.142)$$

Denoting the i^{th} component of $E_{\mathbf{w}}[\mathbf{E}_n]$ as \bar{e}_{ni} , the transfer function between $\mu v_{\text{ss}}\ddot{c}_i(D)$ and \bar{e}_{ni} is

$$t_i(z) = \frac{z^{D_*}}{z^{D_*+1} - z^{D_*} + 2\mu\lambda_i}. \quad (4.143)$$

Then (4.142) converges if and only if, for each i , all of the roots of the characteristic polynomial

$$C(z) = z^{D_*+1} - z^{D_*} + 2\mu\lambda_i \quad (4.144)$$

lie within the unit circle. This is exactly the result obtained in [49] and the bound on μ is found to be

$$0 < \mu < \frac{1}{\lambda_{\max}} \sin \left[\frac{\pi}{2(2D_* + 1)} \right]. \quad (4.145)$$

Using the final value theorem [57], the steady-state value of the i^{th} error vector component is

$$\begin{aligned} \bar{e}_{\text{ssi}} &= \lim_{z \rightarrow 1} \frac{\mu v_{\text{ss}}\ddot{c}_i(D)z^{D_*+1}}{z^{D_*+1} - z^{D_*} + 2\mu\lambda_i} \\ &= \frac{v_{\text{ss}}\ddot{c}_i(D)}{2\lambda_i}, \end{aligned} \quad (4.146)$$

which indicates that there is a bias on the weight vector estimate identical to the one in (4.85). The following proposition characterizes the convergence in the mean of the weight vector.

Proposition 4.6. *In steady-state conditions, the weight vector estimator, given by the adaptive filter LMS algorithm operating jointly with a delay tracking algorithm in Type II-DAB configuration, converges in the mean if*

$$0 < \mu < \frac{1}{\lambda_{\max}} \sin \left[\frac{\pi}{2(2D_* + 1)} \right], \quad (4.147)$$

where λ_{\max} denotes the maximum value of the input signal autocorrelation matrix \mathbf{R} and $D_* = D/T$ is the mean of the delay estimate. The weight vector estimate experiences a bias given by

$$\mathbf{b} = 1/2v_{\text{ss}}\mathbf{R}^{-1}\ddot{\mathbf{p}}(D). \quad (4.148)$$

■

Convergence in the Mean Square of the Weight Vector Estimate when D_* is an Integer

From (4.135), the weight noise vector is

$$\boldsymbol{\eta}_{n+1} = \boldsymbol{\eta}_n + 2\mu e(n, d_n) \mathbf{u}(nT - d_n). \quad (4.149)$$

Using the assumptions and the procedure of Subsection 4.3.1.2, the weight noise correlation matrix $\mathbf{K}_\eta(n+1)$ is found to be similar to (4.98) and is of the form

$$\begin{aligned} \mathbf{K}_\eta(n+1) = & \mathbf{K}_\eta(n) - 2\mu[\mathbf{K}_\eta(n - D_*)\mathbf{R} + \mathbf{R}\mathbf{K}_\eta^T(n - D_*)] + 4\mu^2 \mathbf{R} \text{tr}[\mathbf{R}\mathbf{K}_\eta(n - D_*)] \\ & + 4\mu^2 \mathbf{R}[\xi_{\min} + \ddot{\xi}_{\min} v_{ss}/2] + 4\mu^2 D_* \mathbf{R}[\mathbf{K}_1(n - D_*) + \mathbf{K}_1^T(n - D_*)]\mathbf{R}, \end{aligned} \quad (4.150)$$

where $\mathbf{K}_\eta(n - D_*)$ is defined as

$$\mathbf{K}_\eta(n - D_*) = E[\boldsymbol{\eta}_{nT-d_n} \boldsymbol{\eta}_{nT-d_n}^T], \quad (4.151)$$

which is obtained through an argumentation similar to the one of (4.139) to (4.141), and $\mathbf{K}_{D_*}(n)$ is defined as

$$\mathbf{K}_{D_*}(n) = E[\boldsymbol{\eta}_n \boldsymbol{\eta}_{nT-d_n}^T], \quad (4.152)$$

for D_* an integer. Note in particular, that (4.150) is equal to (4.98) when $D_* = 0$.

Then, using (4.149), the matrix $\mathbf{K}_{D_*}(n)$ is given by

$$\begin{aligned} \mathbf{K}_{D_*}(n) = & E[(\boldsymbol{\eta}_{n-1} + 2\mu e(n-1, d_{n-1}) \mathbf{u}(nT - T - d_{n-1})) \boldsymbol{\eta}_{nT-d_n}^T] \\ = & E[\boldsymbol{\eta}_{n-1} \boldsymbol{\eta}_{nT-d_n}^T] - 2\mu E[\mathbf{u}(nT - T - d_{n-1}) \mathbf{u}^T(nT - T - d_{n-1}) \boldsymbol{\eta}_{nT-T-d_{n-1}} \boldsymbol{\eta}_{nT-d_n}^T] \\ = & \mathbf{K}_{D_*-1}(n-1) - 2\mu \mathbf{R} \mathbf{K}_1^T(n - D_*), \end{aligned} \quad (4.153)$$

where (4.152) is used and the term of the form of (4.89) is neglected for a small delay variance. Applying (4.153) successively, the following result is obtained.

$$\mathbf{K}_{D_*}(n) = \mathbf{K}_\eta(n - D_*) - 2\mu D_* \mathbf{R} \mathbf{K}_1^T(n - D_*). \quad (4.154)$$

Then

$$\begin{aligned} \mathbf{K}_1(n - D_*) = & \mathbf{K}_\eta(n - D_* - 1) - 2\mu \mathbf{R} \mathbf{K}_1^T(n - D_* - 1) \\ = & \mathbf{K}_\eta(n - D_* - 1) - 2\mu \mathbf{R} \mathbf{K}_\eta^T(n - D_* - 2) + 4\mu^2 \mathbf{R}^2 \mathbf{K}_1(n - D_* - 2) \\ = & \sum_{i=0}^{n-D_*-1} (-2\mu \mathbf{R})^i \mathbf{K}_\eta^{Ti}(n - D_* - 1 - i), \end{aligned} \quad (4.155)$$

where

$$\mathbf{K}_\eta^{Ti}(n) = \begin{cases} \mathbf{K}_\eta(n) & \text{for } i \text{ even} \\ \mathbf{K}_\eta^T(n) & \text{for } i \text{ odd.} \end{cases} \quad (4.156)$$

Using the result (4.155) and the definitions of (4.100), (4.101), (4.104) and (4.105) in (4.150) gives the following recursive equation for the diagonal vector $\mathbf{x}(n)$ of the normalized correlation matrix $\mathbf{X}(n)$;

$$\begin{aligned} \mathbf{x}(n+1) = & \mathbf{x}(n) - 4\mu\Lambda\mathbf{x}(n - D_*) + 4\mu^2\lambda\lambda^T\mathbf{x}(n - D_*) \\ & + 4\mu^2\lambda[\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2] + 8\mu^2D_*\Lambda^2 \sum_{i=0}^{n-D_*-1} (-2\mu\Lambda)^i\mathbf{x}(n - D_* - 1 - i). \end{aligned} \quad (4.157)$$

In order to obtain a bound on μ that insures convergence of this equation, it is easier to use (4.157) in the computation of the quantity $\tilde{\xi}(n)$, defined as

$$\tilde{\xi}(n) = E[\eta_n^T \mathbf{R} \eta_n], \quad (4.158)$$

which can be shown to be equal to $\lambda^T \mathbf{x}(n)$ (see equation (4.117)). The quantity $\tilde{\xi}(n)$ is a constituent of the excess MSE (see (4.116)) and must therefore be finite in order to have convergence in the mean square.

In order to simplify the results, assume that the eigenvalues λ_i are nearly equal [†] and that the average eigenvalue is denoted as $\bar{\lambda}$ (this assumption was used with success in [56]). Then, premultiplying both sides of (4.157) by λ^T and using the definition of (4.158) results in

$$\begin{aligned} \tilde{\xi}(n+1) = & \tilde{\xi}(n) - 4\mu\bar{\lambda}\tilde{\xi}(nT - D) + 4\mu^2\text{tr}[\mathbf{R}^2]\tilde{\xi}(nT - D) \\ & + 4\mu^2\text{tr}[\mathbf{R}^2][\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2] + 8\mu^2D_*\bar{\lambda}^2 \sum_{i=0}^{n-D_*-1} (-2\mu\bar{\lambda})^i\tilde{\xi}(n - D_* - 1 - i). \end{aligned} \quad (4.159)$$

Taking the z-transform and rearranging gives

$$\begin{aligned} \tilde{\xi}(z) = & \left[z^{-1} + \left\{ 4\mu^2\text{tr}[\mathbf{R}^2] - 4\mu\bar{\lambda} + \frac{8\mu^2D_*\bar{\lambda}^2}{z + 2\mu\bar{\lambda}} \right\} z^{-D_*-1} \right] \tilde{\xi}(z) \\ & + 4\mu^2\text{tr}[\mathbf{R}^2][\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2] \frac{1}{z-1}. \end{aligned} \quad (4.160)$$

The characteristic polynomial is

$$\begin{aligned} F(z) = & z^{D_*+2} - (1 - 2\mu\bar{\lambda})z^{D_*+1} - 2\mu\bar{\lambda}z^{D_*} \\ & + 4\mu(\bar{\lambda} - \mu\text{tr}[\mathbf{R}^2])z + 8\mu^2\bar{\lambda} \left[\bar{\lambda} - D_*\bar{\lambda} - \mu\text{tr}[\mathbf{R}^2] \right]. \end{aligned} \quad (4.161)$$

In order for (4.159) to be stable, the characteristic polynomial must have all its roots within the unit circle. Jury's test [58] establishes four necessary and sufficient conditions for the characteristic polynomial to have such roots. The first condition is

$$F(1) > 0,$$

[†] This situation is desired in practice to insure reasonable convergence speed of the LMS adaptive filter.

which reduces to

$$2\bar{\lambda}\text{tr}[\mathbf{R}^2]\mu^2 + (2D_*\bar{\lambda}^2 + \text{tr}[\mathbf{R}^2] - 2\bar{\lambda}^2)\mu - \bar{\lambda} < 0. \quad (4.162)$$

This equation represents an upward parabola in μ with a negative minimum. The positive range of values of μ for which the equation is negative is

$$0 < \mu < \frac{\sqrt{(2D_*\bar{\lambda}^2 + \text{tr}[\mathbf{R}^2] - 2\bar{\lambda}^2)^2 + 8\bar{\lambda}^2\text{tr}[\mathbf{R}^2]} - (2D_*\bar{\lambda}^2 + \text{tr}[\mathbf{R}^2] - 2\bar{\lambda}^2)}{4\bar{\lambda}\text{tr}[\mathbf{R}^2]}, \quad (4.163)$$

which is identical to the bound defined in [56]. In this article, it is shown that the second and third conditions of Jury's test hold when this bound is used. The fourth condition cannot be verified analytically, but it is never violated in the simulations performed in [56] and it is therefore conjectured that it is true.

Because $\tilde{\xi}(n) = \boldsymbol{\lambda}^T \mathbf{x}(n)$, the above stability range is also applicable to the convergence of $\mathbf{x}(n)$ given in (4.157). The steady-state value of $\tilde{\xi}(n)$ is obtained by applying the final value theorem to (4.160). For $\mu\bar{\lambda} \ll 1/2$, the result is

$$\begin{aligned} \tilde{\xi}_{ss} &\approx \frac{\mu\text{tr}[\mathbf{R}^2][\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2]}{\bar{\lambda}(1 - 2\mu D_*\bar{\lambda}) - \mu\text{tr}[\mathbf{R}^2]} \\ &\approx \frac{\mu\bar{\lambda}[\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2]}{\bar{\lambda}(1 - 2\mu D_*\bar{\lambda}) - \mu\text{tr}[\mathbf{R}^2]} \boldsymbol{\lambda}^T \mathbf{1}, \end{aligned} \quad (4.164)$$

where $\mathbf{1}$ is an $M \times 1$ unit vector, i.e. it has all its elements equal to 1 and the second equation is obtained by assuming nearly equal eigenvalues. Then

$$\mathbf{x}_{ss} \approx \frac{\mu\bar{\lambda}[\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2]}{\bar{\lambda}(1 - 2\mu D_*\bar{\lambda}) - \mu\text{tr}[\mathbf{R}^2]} \mathbf{1}. \quad (4.165)$$

The convergence in the mean square is therefore formalized in the following proposition.

Proposition 4.7. *In steady-state conditions, the weight vector estimator, given by the adaptive filter LMS algorithm operating jointly with a mean square convergent integer delay tracking algorithm in Type II-DAB configuration, is convergent in the mean square if*

$$0 < \mu < \frac{\sqrt{(2D_*\bar{\lambda}^2 + \text{tr}[\mathbf{R}^2] - 2\bar{\lambda}^2)^2 + 8\bar{\lambda}^2\text{tr}[\mathbf{R}^2]} - (2D_*\bar{\lambda}^2 + \text{tr}[\mathbf{R}^2] - 2\bar{\lambda}^2)}{4\bar{\lambda}\text{tr}[\mathbf{R}^2]}, \quad (4.166)$$

where λ_i is the i^{th} eigenvalue of the $M \times M$ input signal autocorrelation matrix \mathbf{R} , $\bar{\lambda}$ is the average eigenvalue and $D_* = D/T$ is the mean of the delay estimator. ■

From (4.165), the trace of the correlation matrix is

$$\begin{aligned} \text{tr}[\mathbf{K}_\eta] &= \sum_{i=1}^M x_{i(ss)} \\ &\approx \mu M \frac{\bar{\lambda}[\xi_{\min} + \ddot{\xi}_{\min}v_{ss}/2]}{\bar{\lambda}(1 - 2\mu D_*\bar{\lambda}) - \mu\text{tr}[\mathbf{R}^2]}. \end{aligned} \quad (4.167)$$

Note that if $D_* = 0$, (4.167) reduces to (4.111), when the eigenvalues are nearly equal.

Approximation of $\phi''_{\eta_i \eta_i}(0)$

In order to compute the approximation of (4.133) for $k = 1$, the diagonal elements of the cross-correlation matrix $\mathbf{K}_1(n)$ must be available. From (4.155), this matrix is given by

$$\begin{aligned} \mathbf{K}_1(n) &= E[\boldsymbol{\eta}_n \boldsymbol{\eta}_{n-1}] \\ &= \sum_{i=0}^{n-1} (-2\mu \mathbf{R})^i \mathbf{K}_\eta^{T^i}(n-1-i), \end{aligned} \quad (4.168)$$

from which the normalized diagonal vector can be obtained. It is given by

$$\mathbf{x}_1(n) = \sum_{i=0}^{n-1} (-2\mu \Lambda)^i \mathbf{x}(n-1-i), \quad (4.169)$$

where

$$\mathbf{x}_1(n) = \text{diag}[\mathbf{Q}^T \mathbf{K}_1(n) \mathbf{Q}], \quad (4.170)$$

and $\mathbf{x}(n)$ is defined in (4.104). The j^{th} component of $\mathbf{x}_1(n)$ can be expressed as (using (4.154) with $D_* = 1$)

$$x_{1j}(n) = x_j(n-1) - 2\mu \lambda_j x_{1j}(n-1). \quad (4.171)$$

Assuming that the conditions of convergence are respected, the steady-state value of $x_{1j}(n)$ is

$$x_{1j}(\infty) = \frac{x_j(\infty)}{1 + 2\mu \lambda_j}, \quad (4.172)$$

which is approximately equal to $x_j(\infty)$ when the condition of (4.134) is respected. Therefore, $\phi_{\eta_i \eta_i}(1)$ is approximately equal to $\phi_{\eta_i \eta_i}(0)$, and $\phi''_{\eta_i \eta_i}(0)$ is approximately zero.

4.3.2.3 Excess Mean-Squared Error and Misadjustment with the Joint LMS Algorithm in Type II-DAB Configuration

Proceeding as in Subsection 4.3.1.3, the MSE function is

$$\xi_{\text{ss}} = \xi_{\text{min}} + v_{\text{ss}} \ddot{\xi}_{\text{min}}/2 + E[\boldsymbol{\eta}_{nT-d_n}^T \mathbf{R} \boldsymbol{\eta}_{nT-d_n}]. \quad (4.173)$$

From (4.117) and (4.164), the last term of (4.173) is given by

$$\begin{aligned} E[\boldsymbol{\eta}_{nT-d_n}^T \mathbf{R} \boldsymbol{\eta}_{nT-d_n}] &= \text{tr}[\mathbf{R} \mathbf{K}_\eta(\infty)] \\ &= \boldsymbol{\lambda}^T \mathbf{x}_{\text{ss}} \\ &= \tilde{\xi}_{\text{ss}} \\ &\approx \frac{\mu \text{tr}[\mathbf{R}^2][\xi_{\text{min}} + \ddot{\xi}_{\text{min}} v_{\text{ss}}/2]}{\bar{\lambda}(1 - 2\mu D_* \bar{\lambda}) - \mu \text{tr}[\mathbf{R}^2]}. \end{aligned} \quad (4.174)$$

Therefore, the excess MSE has the same form as for the Type I configuration and is given by

$$\xi_{\text{ex}} = \xi_{\text{ex}}^{\text{d}} + \xi_{\text{ex}}^{\text{f}} + \xi_{\text{ex}}^{\text{df}}, \quad (4.175)$$

where

$$\xi_{\text{ex}}^{\text{d}} = \frac{v_{\text{ss}} \ddot{\xi}_{\text{min}}}{2}, \quad (4.176)$$

$$\xi_{\text{ex}}^{\text{f}} = \frac{\mu \xi_{\text{min}} \text{tr}[\mathbf{R}^2]}{\bar{\lambda}(1 - 2\mu D_* \bar{\lambda}) - \mu \text{tr}[\mathbf{R}^2]}, \quad (4.177)$$

$$\xi_{\text{ex}}^{\text{df}} = \frac{\mu \ddot{\xi}_{\text{min}} v_{\text{ss}} \text{tr}[\mathbf{R}^2]}{2(\bar{\lambda}(1 - 2\mu D_* \bar{\lambda}) - \mu \text{tr}[\mathbf{R}^2])}. \quad (4.178)$$

The misadjustment has the form

$$\begin{aligned} \mathcal{M} &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{df}} \\ &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{d}} \mathcal{M}^{\text{f}}, \end{aligned} \quad (4.179)$$

where the different terms are trivially related to the corresponding excess MSE terms of (4.175) to (4.178).

4.3.3 The Joint LMS Algorithm in Type II Configuration: Delay in Reference Branch

A Type II-DRB system in cancellation mode is illustrated in Figure 3.2 of Chapter 3. This type of configuration simplifies considerably the analysis of the Type II system and makes it more practical since it avoids the delay between the filter adaptation and the error signal. The negative delay $-d_n$ is implemented in practice by introducing a *fixed* delay before the adaptive filter. The error signal is given in (4.5) and the corresponding joint LMS algorithm is

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n + 2\mu e(n, d_n) \mathbf{u}(nT) \\ d_{n+1} &= d_n - 2\alpha e(n, d_n) \hat{r}(nT - d_n). \end{aligned} \quad (4.180)$$

4.3.3.1 Analysis for the LMS Delay Estimator in Steady-State

Because of the adaptive delay location, the output of the adaptive filter is independent of d_n . But the optimum adaptive filter output $\hat{r}(n)$ is still a function of $d_n = D$ and the noisy output is defined as

$$y(n, D) = \hat{r}(n) + \chi(n, D), \quad (4.181)$$

where

$$\hat{r}(n) = \mathbf{w}_{\text{opt}}^T \mathbf{u}(nT) \quad (4.182)$$

and

$$\chi(n, D) = \boldsymbol{\eta}_n^T \mathbf{u}(nT) |_{d_n = -D}. \quad (4.183)$$

The Taylor approximation of $\gamma_n(d_n) = e^2(n, d_n)$ is still used with

$$\begin{aligned} \dot{\gamma}_n(D) &= 2e(n, D)\dot{r}(nT + D) \\ \ddot{\gamma}_n(D) &= 2\dot{r}^2(nT + D) + 2e(n, D)\ddot{r}(nT + D). \end{aligned} \quad (4.184)$$

Defining

$$N_n = 2e(n, D)\dot{r}(nT + D) \quad (4.185)$$

and

$$G_n = \dot{r}^2(nT + D) + e(n, D)\ddot{r}(nT + D), \quad (4.186)$$

the approximate LMS delay tracking algorithm is (compare to equation (4.47))

$$d_{n+1} = d_n - 2\alpha(d_n + D)G_n - \alpha N_n. \quad (4.187)$$

Convergence in the Mean of the Delay Estimate

Proceed as in Subsection 4.3.1.1, i.e. take the expected value of (4.187). Note that Proposition 4.1 holds and that

$$\begin{aligned} E[G_n] &= -\phi''_{\dot{r}\dot{r}}(D) \\ &= 1/2\ddot{\xi}_{\min} \end{aligned} \quad (4.188)$$

$$E[N_n] = 0,$$

as in (4.51) and (4.57). Then Proposition 4.2 applies in the present case, i.e. the condition of convergence in the mean is

$$0 < \alpha < \frac{2}{\ddot{\xi}_{\min}}. \quad (4.189)$$

Convergence in the Mean Square of the Delay Estimate

Apply the procedure of Subsection 4.3.1.1. Note that $E[(1 - 2\alpha G_n)N_n] = 0$ again. Then the same mean square analysis applies and Proposition 4.3 is valid with

$$\begin{aligned} E[G_n^2] &= 3(\phi''_{\dot{r}\dot{r}}(0))^2 - 2\phi''_{\dot{r}\dot{r}}(0)[\phi''_{\dot{r}\dot{r}}(0) - \phi''_{\dot{r}\dot{r}}(D)] + 2[\phi''_{\dot{r}\dot{r}}(0) - \phi''_{\dot{r}\dot{r}}(D)]^2 \\ &\quad + (\phi_{\dot{r}\dot{r}}(0) - \phi_{\dot{r}\dot{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])\phi_{\dot{r}\dot{r}}^{(4)}(0). \end{aligned} \quad (4.190)$$

The steady-state delay estimate variance is still given by (see (4.74))

$$v_{\text{ss}} = \frac{\alpha E[N_n^2]}{2\ddot{\xi}_{\min} - 4\alpha E[G_n^2]}, \quad (4.191)$$

where $E[N_n^2]$ can be shown to be

$$E[N_n^2] = -4(\phi_{\dot{r}\dot{r}}(0) - \phi_{\dot{r}\dot{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])\phi''_{\dot{r}\dot{r}}(0). \quad (4.192)$$

4.3.3.2 Analysis for the LMS Adaptive Filter in Steady-State

Combining the first equation of (4.180) and the error definition of (4.5), the LMS adaptive filter algorithm is

$$\mathbf{w}_{n+1} = \mathbf{w}_n + 2\mu[r(nT - d_n)\mathbf{u}(nT) - \mathbf{u}(nT)\mathbf{u}^T(nT)\mathbf{w}_n]. \quad (4.193)$$

The mean and mean square analyses, based on (4.193), give the same results as those of Subsection 4.3.1.2, with D replaced by $-D$, and Propositions 4.4 and 4.5 are valid in the present case.

4.3.3.3 Excess Mean-Squared Error and Misadjustment with the Joint LMS Algorithm

The results of Subsection 4.3.1.3 apply integrally, with the obvious changes in $E[G_n^2]$ and $[N_n^2]$ according to (4.190) and (4.192) (for the computation of v_{ss}).

4.4 Discussion

As pointed out in Chapter 3, the joint steepest-descent algorithm and its stochastic counterpart, the joint LMS algorithm, represent the generalizations of either the conventional SD (LMS) delay tracking algorithm [29] or the conventional SD (LMS) adaptive transversal filter algorithm [10]. It is therefore not surprising to find that all the results of Subsections 4.3.1.1, 4.3.2.1 and 4.3.3.1, about the delay algorithm, degenerate to the results of [29] when the signals are properly interpreted, and that the results of Subsections 4.3.1.2, 4.3.2.2 and 4.3.3.2 come down to the LMS adaptive filter results, when the delay D and the variance are set equal to zero.

Another point to note is the fact that, as long as the delay estimation algorithm is convergent in the mean square (v_{ss} is finite), the conditions for convergence of the LMS adaptive filter, in the mean and in the mean square, are identical to the usual conditions for a similar adaptive filter operating alone or with a fixed delay element, i.e. the convergence depends on the eigenvalues of the input signal autocorrelation matrix. Note also that, because of the adaptive delay element, the weight vector estimate is biased.

As equations (4.73) and (4.74) suggest it, the convergence of the LMS adaptive delay element depends on ξ_{\min} , $E[G_n^2]$ and $E[N_n^2]$, for the three types of systems. Using (4.57) and the fact that

$$\xi_{\min} = \phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0), \quad (4.194)$$

equations (4.69) and (4.75) can take the form † (Types I and II-DAB)

$$\begin{aligned}
E[G_n^2] &\approx 3/4\ddot{\xi}_{\min}^2 - 1/2\xi_{\min}\xi_{\min}^{[4]} \\
&\quad + [\xi_{\min}\phi_{uu}^{(4)}(0) - 1/2\xi_{\min}^{[4]}\phi_{uu}(0) - 2\ddot{\xi}_{\min}\phi_{uu}''(0)]\text{tr}[\mathbf{K}_\eta] \\
&\quad + [3(\phi_{uu}''(0))^2 + \phi_{uu}(0)\phi_{uu}^{(4)}(0)]\text{tr}^2[\mathbf{K}_\eta]
\end{aligned} \tag{4.195}$$

$$\begin{aligned}
E[N_n^2] &\approx 2\xi_{\min}\ddot{\xi}_{\min} \\
&\quad + [2\ddot{\xi}_{\min}\phi_{uu}(0) - 4\xi_{\min}\phi_{uu}''(0)]\text{tr}[\mathbf{K}_\eta] \\
&\quad - 4\phi_{uu}(0)\phi_{uu}''(0)\text{tr}^2[\mathbf{K}_\eta],
\end{aligned} \tag{4.196}$$

and equations (4.190) and (4.192) become (Type II-DRB)

$$\begin{aligned}
E[G_n^2] &\approx 3(\phi_{rr}''(0))^2 - 2\phi_{rr}''(0)[\phi_{rr}''(0) + 1/2\ddot{\xi}_{\min}] + 2[\phi_{rr}''(0) + 1/2\ddot{\xi}_{\min}]^2 \\
&\quad + (\xi_{\min} + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])\phi_{rr}^{(4)}(0)
\end{aligned} \tag{4.197}$$

$$E[N_n^2] \approx -4(\xi_{\min} + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])\phi_{rr}''(0). \tag{4.198}$$

Equations (4.195) to (4.198) indicate that the convergence of the LMS adaptive delay element depends on the input signal power $\phi_{uu}(0)$ and the minimum MSE ξ_{\min} in the Types I and II-DAB, as well as on the reference signal power $\phi_{rr}(0)$ in the Type II-DRB case.

The expression (4.74) (valid for the three types of systems) for the delay estimate variance is complicated by the presence of the adaptive filter-related terms. The delay estimate variance is also encountered in the excess MSE and misadjustment expressions, like (4.121) and (4.125). Once the delay variance is computed or fixed, these two quantities are seen to be functions of two terms specific to the adaptive delay element and to the adaptive filter, respectively, and of a cross-product term (note that the delay specific term being function of v_{ss} , it is indirectly function of the adaptive filter). Note that the expressions for $\xi_{\text{ex}}^{\text{d}}$ and $\xi_{\text{ex}}^{\text{f}}$ are identical to those obtained for the respective adaptive algorithms operating alone [29], [7]. The cross-product terms $\xi_{\text{ex}}^{\text{df}}$ and M^{df} are essentially the result of gradient and derivative estimation noise in the two adaptation processes. For stationary input and reference processes, the estimation noise in one adaptive algorithm is increased by the gradient estimation noise present in the other adaptive system. Therefore, the total misadjustment \mathcal{M} is not merely the sum of the adaptive delay element and adaptive filter misadjustment expressions \mathcal{M}^{d} and \mathcal{M}^{f} , but also includes a term due to the joint estimation noise. Note, however, that the cross-product misadjustment \mathcal{M}^{df} is equal to the product of \mathcal{M}^{d} and \mathcal{M}^{f} , which makes it a second-order term that can be, in practical situations, one order of magnitude smaller than the individual terms.

† Note that these expressions are *exact* for white input and noise signals.

The results obtained in this chapter are based on a number of assumptions, as listed at the beginning of Section 4.3. These assumptions may seem restrictive, but they can be justified as follows. The Gaussian assumption is a common one and has been used in most of the more involved analyses, as in [59], [60] or [7]. The whiteness assumption in the noise processes is more specific, but it is often met in practice and is used only in the proof of Proposition 4.1. Assumption 2 about the stationarity of the reference signal is used to limit the analysis to the effects of the gradient and derivative noises on the steady-state behaviour of the joint algorithm. The excess MSE and misadjustment caused by the tracking lag, in the case of nonstationary reference signals, was therefore not considered in the analysis. The independence assumptions 3 and 5 are also common in the analysis of stochastic algorithms. The zero-mean Gaussian assumption about the weight noise vector (Assumption 5), when the adaptive delay element is considered, is clearly wrong in view of the bias in the adaptive noise vector (see Proposition 4.4). But practical considerations ask for a small delay variance, in which case the weight vector bias is also small and Assumption 5 almost valid. Finally, the assumption of high signal-to-noise ratio is used, as in Chapter 3, to simplify the results and obtain useful indications about the algorithm.

4.5 Application of the Joint LMS Algorithm

The application of the various results obtained in this chapter is not an obvious task, due mainly to the complexity of the different formulas and to the relationships among them. But as shown above, the different bounds are functions of the input and reference signals, and can therefore be estimated.

Note that if μ and ξ_{\min} are small, the quantity $\text{tr}[\mathbf{K}_\eta]$ is approximately zero and $E[G_n^2] \approx 3/4\xi_{\min}^2$ for a Type I system (see (4.195)). In this case, convergence in the mean square happens for

$$0 < \alpha < \frac{2}{3\xi_{\min}},$$

which is $1/3$ of the upper bound for convergence in the mean (see Proposition 4.2).

In order to use the convergence bounds on α and μ , it is necessary to know the delay estimate variance v_{ss} , which itself is a function of α . Since, in practice, a certain variance is desired or desirable, v_{ss} can be used as a design variable that is fixed *a priori*. The different quantities which are functions of this variance are then computed more easily.

A Type I system design procedure, for the determination of α and μ in high signal-to-noise conditions, can take the following form

1. Assume an acceptable delay steady-state variance v_{ss} .
2. Estimate $\ddot{\xi}_{\min}$, $\phi_{uu}(0)$ and its derivatives (proceed as in Section 3.4.1, in particular equation (3.117)).
3. Compute $\text{tr}[\mathbf{K}_\eta]$, $E[G_n^2]$ and $E[N_n^2]$, as functions of μ , using equations (4.111), (4.195) and (4.196).
4. Obtain a relationship between α and μ , using equation (4.74).
5. Use equation (3.91) of Chapter 3 to get a second relationship between α and μ and solve for these two factors.
6. Verify that the convergence bounds, for both α and μ , are satisfied.

Similar procedures can be described for the two other types of systems. Because of the assumptions used, these design procedures are useful only if they are used with caution to obtain approximate information about the algorithms. More results concerning the applicability of the procedures are given in Chapter 6.

The different bounds developed in Chapter 3 are useful in the application of the joint LMS algorithm. In particular, note that the conditions for convergence in the mean of the delay estimator in Type I or Type II-DRB (equation (4.60)) is the same as the stability range for the SD delay estimator (equation (3.75)). Then the tighter bounds of Proposition 3.2 (equations (3.82) and (3.83)) can be used to predict the convergence in the mean of the delay estimator. The other results of Subsection 3.3.2.2 can also be use with profit in the application of the joint LMS algorithm.

Finally, note that the analysis and the results obtained for the Type II-DAB adaptive system (Subsection 4.3.2) are the least appealing and realistic ones. These results should mainly be considered as indicative of the fact that a Type II-DRB configuration is more attractive and should be preferred. Nevertheless, practical situations may dictate the choice of a Type II-DAB form, in which case the theoretical results could be of interest.

4.6 Summary

Joint time delay estimation and adaptive MMSE filtering, using the least-mean-square algorithm, has been studied in details in this chapter. The differences between three Types of joint algorithms (I, II-DAB and II-DRB) were established, and in the Type I case, it was assumed that the delay d_n propagates instantaneously into the adaptive filter delay line. The ODE method was used to show that when the adaptation factors α and μ both tend toward zero, the joint LMS algorithm converges to a local minimum of the MSE function,

like the exact version of the joint steepest-descent algorithm. This supports the fact that, when the factors are small, the joint LMS algorithm converges to solutions close to those of the joint SD algorithm.

The three types of joint LMS algorithm were studied in steady-state conditions, when the reference signal is stationary. It was established that the adaptive delay element convergence bounds are governed by the input signal power and the second derivative of the MSE function at its minimum in a Type I system, and by the same quantities, plus the reference signal power, in the Type II-DRB case. In these two types, the adaptive filter convergence bounds were found to be given by expressions identical to those obtained for an adaptive filter operating alone. It was also found that the delay estimate is unbiased, while the weight vector estimate is biased by a quantity proportional to the delay estimate variance. It was also argued that a Type II-DRB adaptive system should be preferred to a Type II-DAB system. A design procedure for the choice of the adaptation factors was discussed, and it was pointed out that the results of Subsection 3.3.2.2 could be used with profit, in the application of the joint LMS algorithm.

The material presented in this chapter shows explicitly the complexity of the analysis of stochastic joint algorithms, and could be seen as an attempt to unify the analyses of LMS adaptive delay and adaptive filter algorithms, as well as a unification of the analyses of different types of joint LMS delay estimation and adaptive filtering algorithms.

system acts as if the other system was not present). This philosophy can be applied to any type of adaptive configuration, as defined in Chapter 3 (Type I, II-DAB or II-DRB). But the particularity of the RLS adaptive filter algorithm is that it computes the true solution of the LS problem at each iteration, which typically insures a rate of convergence an order of magnitude faster than the simple SD or LMS algorithms [7]. This characteristic can prevent the use of an independent delay estimation algorithm, as in the joint MMSE algorithm of Chapter 3. This is so because the adaptive filter converges so quickly that it will model by itself the most part of any reference delay before the adaptive delay loop can converge. In most occasions, the joint LS algorithm must therefore intimately link the two adaptive processes.

Another problem with the use of the RLS adaptive filter algorithm is its inherent computational complexity (the LS solution involves in fact the inversion of the input signal autocorrelation matrix). The use of a fractional delay element involves an additional complexity that is not welcome.

These problems can be partially circumvented by using an *integer* delay element that is not updated only in the direction of the least squares solution, as in a gradient-type algorithm, but that selects a value that truly minimizes $\mathcal{E}(n)$ at each iteration, within a finite set of possible delay values. This type of joint algorithm computes the two estimates such that they correspond to the joint LS solution at each iteration.

In this chapter, two new joint delay and reference filter tracking algorithms of this kind are proposed. One is based on the Type I configuration (the adaptive delay is located before the adaptive filter) and the other assumes a Type II-DRB adaptive system (the adaptive delay is located in the reference branch). Define the integer time delay as a time *lag* and denote it by ℓ . Then, the error $e(i, d_i)$ in (5.1) can be expressed as

$$\begin{aligned} e(i, d_i) &= e(i, \ell) \\ &= r(i) - \mathbf{w}^H(n)\mathbf{u}(i - \ell) \quad \text{Type I,} \\ &= r(i + \ell) - \mathbf{w}^H(n)\mathbf{u}(i) \quad \text{Type II - DRB.} \end{aligned} \tag{5.2}$$

For an adaptive filter with a given number of taps M , define the minimum sum of weighted squared errors $\hat{\xi}_M(n)$ as

$$\hat{\xi}_M(n) = \min_{\mathbf{w}(n), \ell} \mathcal{E}(n), \tag{5.3}$$

where the minimization with respect to ℓ is accomplished over a finite set of lag values. Then, for a given value of ℓ , define the minimum sum $\hat{\xi}_{M_o}(n, \ell)$ as (compare with the definition of $\xi_o(d_n)$ in equation (3.39))

$$\hat{\xi}_{M_o}(n, \ell) = \min_{\mathbf{w}(n)} \mathcal{E}(n). \tag{5.4}$$

The weight vector for which this minimum is attained is defined as $\hat{\mathbf{w}}_M^\ell(n)$. If the adaptive delay d_i is not equal to the reference delay D_i , for all i , the sum of errors $\hat{\xi}_{M_o}(n, d_n)$ is not minimum with respect to d_n , unless the adaptive filter length is large enough to accommodate both the modelling of the reference filter $h(n)$ and the reference delay (i.e. M is large enough such that the delayed optimum adaptive weight vector is not truncated).

The RLS algorithms derived in this chapter exploit the data structure in order to compute the adaptive weight vector and the lag value, within a finite set, corresponding to the joint LS solution. In order to perform such a task, the sum of squared errors $\hat{\xi}_{M_o}(n, \ell)$ is computed for each value of ℓ in the set of interest, and the delay value corresponding to the lowest value is retained. The set of possible delay values is chosen to be $\{\ell - 1, \ell, \ell + 1\}$.

The joint LS lag estimation and adaptive filtering algorithms can be cast into the following general algorithmic form

1. Apply the Recursive Least Squares (RLS) algorithm in order to obtain $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{\xi}_{M_o}(n, \ell)$
2. Adapt ℓ by using derivative information from $\hat{\xi}_{M_o}(n, \ell)$ and update $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{\xi}_{M_o}(n, \ell)$.

Conceptually, the first part of the algorithm can be implemented by using any of the computationally efficient forms of the RLS algorithm, and the second part can be implemented as a gradient search, with respect to ℓ , of $\hat{\xi}_{M_o}(n, \ell)$. The gradient can be given, for example, by

$$\frac{\partial \hat{\xi}_{M_o}(n, \ell)}{\partial \ell} = \begin{cases} 1 & \text{if } \hat{\xi}_{M_o}(n, \ell + 1) < \hat{\xi}_{M_o}(n, \ell) \text{ and } \hat{\xi}_{M_o}(n, \ell + 1) < \hat{\xi}_{M_o}(n, \ell - 1) \\ -1 & \text{if } \hat{\xi}_{M_o}(n, \ell - 1) < \hat{\xi}_{M_o}(n, \ell) \text{ and } \hat{\xi}_{M_o}(n, \ell - 1) < \hat{\xi}_{M_o}(n, \ell + 1) \\ 0 & \text{otherwise,} \end{cases} \quad (5.5)$$

and the lag value updated as

$$\ell = \ell + \varpi \left\langle \frac{\partial \hat{\xi}_{M_o}(n, \ell)}{\partial \ell} \right\rangle, \quad (5.6)$$

where $\langle \cdot \rangle$ denotes a form of time average and ϖ is a positive constant [†].

[†] The constant ϖ is taken to be equal to one in the rest of the thesis. It is explicitly shown in the lag-update equation in order to relate this equation to the SD delay adaptation algorithm of the previous chapters.

Another form of lag update can rely on a time average of the sum of squared errors, i.e. the derivative can be implemented as

$$\frac{\partial \hat{\xi}_{M_o}(n, \ell)}{\partial \ell} = \begin{cases} 1 & \text{if } \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle \text{ and } \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle \\ -1 & \text{if } \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle \text{ and } \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle \\ 0 & \text{otherwise,} \end{cases} \quad (5.7)$$

and the lag value updated as

$$\ell = \ell + \varpi \frac{\partial \hat{\xi}_{M_o}(n, \ell)}{\partial \ell}. \quad (5.8)$$

This form of joint RLS algorithm is significantly different from the joint LMS and SD algorithms, since it relies on the ability of the adaptive filter to model a delay. The *integer* delay (lag) estimation is performed by extracting the time shift information from the adaptive filter, in order to keep it “centered” to the nearest sample. The fractional part of the reference delay is still modelled by the adaptive filter. Note that ℓ does not carry a time index because, in the RLS algorithm, it is assumed that the signals are stationary within the memory of the algorithm (defined by β), which implies that ℓ applies to all the previous data. Note also that when ℓ is updated, $\hat{\mathbf{w}}_M^\ell(n)$ must also be corrected, in order to obtain the joint solution of (5.3).

In order to compute (5.5) or (5.7), the optimum weight vectors for lags $\ell + 1$ and $\ell - 1$ must be available. This extra information can be obtained by computing the RLS algorithm two more times, in a parallel fashion. This implies an increase in both the computation count and in the storage requirement. Another method of doing the same thing consists in applying the RLS algorithm once, and in deriving the extra information from this single application. This method is made possible by using a set of lag-recursive relations, for the two types of adaptive system considered in (5.2), that allow the *exact* computation of $\hat{\xi}_{M_o}(n, \ell + 1)$, $\hat{\xi}_{M_o}(n, \ell - 1)$, $\hat{\mathbf{w}}_M^{\ell+1}(n)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n)$ from the knowledge of $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{\xi}_{M_o}(n, \ell)$. These lag-recursive relations are derived in this chapter as functions of variables encountered in the different forms of fast transversal LS adaptive filters [62], [61], and are *naturally* appended to these algorithms. The original form of the lag-recursive relations was derived by Kalouptsidis et al. [63] and is extended in the next sections.

The main contributions of this chapter are twofold. Firstly, a new geometrical derivation of the lag-recursive equations, for both $\hat{\xi}_{M_o}(n, \ell)$ and $\hat{\mathbf{w}}_M^\ell(n)$, is performed in Section 5.3. The relations derived in [63] are based on a fixed block of data, while their *on line* counterpart was first presented in [42]. The second contribution is the description of a new joint time delay estimation and adaptive RLS filter, in Section 5.4. The effects of the delay estimation on the RLS algorithm, in steady-state conditions, are considered in Section 5.5.

Finally, note that every explicit derivation presented in this chapter is for a Type II-DRB adaptive system configuration in cancellation mode, of the form of Figure 3.2. The reason for this fact is that the Type II-DRB system is the most practical of the two forms. An integer-value adaptive delay element before the adaptive filter (as in Type I) implies that the whole set of RLS filter recursions is function of ℓ (for a list of these recursions, see Appendix F), and that this entire set has to be updated in the case of lag update. This increases considerably the algorithm computational complexity. In practice, it is preferable to assume that a slowly varying reference delay is present in the reference branch and to use a Type II-DRB adaptive system in all cases. The lag-update relations for a Type I adaptive system will be given and discussed, but they are not the main focus of the chapter.

5.2 Background Theory

In this section, some definitions and notational conventions are presented, along with some geometrical considerations. This background material is used, in the subsequent sections, to derive the lag-recursive relations and to link them to existing fast transversal filter (FTF) algorithms. Some shift invariance properties and common recursions used in the RLS algorithm are discussed in Appendix E. The FTF algorithm that will be considered is discussed in Appendix F.

5.2.1 Notation and Definitions for a Type II-DRB Configuration

In the *prewindowed weighted recursive least squares* adaptation algorithm for adaptive transversal filters of order M , the index of performance to be minimized, at iteration n , and for a lag ℓ in the reference data, is

$$\mathcal{E}(n) = \sum_{i=1}^n \beta^{n-i} |e_M(i, \ell)|^2, \quad (5.9)$$

where the *a posteriori* estimation error is defined by

$$e_M(i, \ell) = r(i + \ell) - \mathbf{w}_M^{\ell H}(n) \mathbf{u}_M(i), \quad (5.10)$$

with

$$\begin{aligned} \mathbf{u}_M(i) &= [u(i), u(i-1), \dots, u(i-M+1)]^T \\ \mathbf{w}_M^\ell(n) &= [w_{1M}^\ell(n), w_{2M}^\ell(n), \dots, w_{MM}^\ell(n)]^T. \end{aligned} \quad (5.11)$$

Note that the prewindowed method assumes that the data is zero prior to iteration $n = 1$ [7]. Define also the *a priori* estimation error $\alpha_M(i, \ell)$ as

$$\alpha_M(i, \ell) = r(i + \ell) - \mathbf{w}_M^{\ell H}(n-1) \mathbf{u}_M(i). \quad (5.12)$$

Another set of vectors can be defined in the complex vector space \mathbf{C}^n of order n . The n -vectors $\mathbf{U}(n)$, $\mathbf{D}^\ell(n)$ and $\mathbf{E}_M^\ell(n)$ are defined as

$$\mathbf{U}(n) = [u(n), u(n-1), \dots, u(1)]^T \quad (5.13)$$

$$\mathbf{D}^\ell(n) = [r(n+\ell), r(n+\ell-1), \dots, r(\ell+1)]^T \quad (5.14)$$

$$\mathbf{E}_M^\ell(n) = [e_M(n, \ell), e_M(n-1, \ell), \dots, e_M(1, \ell)]^T. \quad (5.15)$$

The vector shift operator z^{-j} is defined in \mathbf{C}^n by

$$z^{-j}\mathbf{U}(n) = [u(n-j), u(n-j-1), \dots, u(1), 0, \dots, 0]^T \in \mathbf{C}^n. \quad (5.16)$$

Then, the matrix $\mathbf{A}(n|i, M)$ is defined as

$$\mathbf{A}(n|i, M) = [z^{-i}\mathbf{U}(n), z^{-i-1}\mathbf{U}(n), \dots, z^{-M}\mathbf{U}(n)], \quad (5.17)$$

and the vector subspace spanned by the columns of $\mathbf{A}(n|i, M)$ as $S(n|i, M)$.

The deterministic autocorrelation matrix is defined as (using the notation in [7])

$$\Phi_M(n) = \sum_{i=1}^n \beta^{n-i} \mathbf{u}_M(i) \mathbf{u}_M^H(i), \quad (5.18)$$

and the deterministic cross-correlation vector with lag ℓ as

$$\theta_M^\ell(n) = \sum_{i=1}^n \beta^{n-i} \mathbf{u}_M(i) r^*(i+\ell). \quad (5.19)$$

The least squares weight vector at iteration n , for lag ℓ , is

$$\hat{\mathbf{w}}_M^\ell(n) = \Phi_M^{-1}(n) \theta_M^\ell(n), \quad (5.20)$$

and the corresponding minimum of squared errors is

$$\hat{\xi}_{M0}(n, \ell) = \min_{\mathbf{w}} \mathcal{E}(n) = \sum_{i=1}^n \beta^{n-i} |r(i+\ell) - \hat{\mathbf{w}}_M^{\ell H}(n) \mathbf{u}_M(i)|^2. \quad (5.21)$$

Note that the data is assumed such that the deterministic autocorrelation matrix is non-singular.

Denote the optimum weight vector for the one-step forward linear predictor of order m as $\mathbf{a}_m(n)$. This vector minimizes the sum of weighted forward *a posteriori* prediction-error squares, defined as

$$F_m(n) = \sum_{i=1}^n \beta^{n-i} |f_m(i)|^2, \quad (5.22)$$

where

$$f_m(i) = u(i) - \mathbf{a}_m^H(n) \mathbf{u}_m(i-1). \quad (5.23)$$

The forward *a priori* prediction-error $\eta_m(i)$ is defined as

$$\eta_m(i) = u(i) - \mathbf{a}_m^H(n-1) \mathbf{u}_m(i-1). \quad (5.24)$$

Similarly, the optimum weight vector for the one-step backward linear predictor of order m is the vector $\mathbf{b}_m(n)$ that minimizes the sum of weighted backward *a posteriori* prediction-error squares, defined as

$$B_m(n) = \sum_{i=1}^n \beta^{n-i} |b_m(i)|^2, \quad (5.25)$$

with

$$b_m(i) = u(i-m) - \mathbf{b}_m^H(n) \mathbf{u}_m(i). \quad (5.26)$$

Then the backward *a priori* prediction-error $\psi_m(i)$ is defined as

$$\psi_m(i) = u(i-m) - \mathbf{b}_m^H(n-1) \mathbf{u}_m(i). \quad (5.27)$$

Define the vectors $\mathbf{E}_{M-1}^f(n)$ and $\mathbf{E}_{M-1}^b(n)$ as

$$\mathbf{E}_{M-1}^f(n) = [f_{M-1}(n), f_{M-1}(n-1), \dots, f_{M-1}(1)]^T \quad (5.28)$$

$$\mathbf{E}_{M-1}^b(n) = [b_{M-1}(n), b_{M-1}(n-1), \dots, b_{M-1}(1)]^T. \quad (5.29)$$

5.2.1.1 Shift Invariance Properties

In a geometrical framework, it is noted that the subspace $S(n|0, M-1)$ can be expressed either as

$$S(n|0, M-1) = S(n|1, M-1) \oplus \mathbf{U}(n), \quad (5.30)$$

or as

$$S(n|0, M-1) = S(n|0, M-2) \oplus z^{-M+1} \mathbf{U}(n), \quad (5.31)$$

where the operation \oplus stands for the direct sum operation. Note also that

$$\begin{aligned} S(n|1, M-1) &= \text{span}\{z^{-1} \mathbf{U}(n), z^{-2} \mathbf{U}(n), \dots, z^{-M+1} \mathbf{U}(n)\} \\ &= \text{span}\{\mathbf{U}(n-1), z^{-1} \mathbf{U}(n-1), \dots, z^{-M+2} \mathbf{U}(n-1)\} \\ &= S(n-1|0, M-2) \end{aligned} \quad (5.32)$$

and that

$$\begin{aligned} \mathbf{D}^\ell(n) &= [r(n-1+\ell+1), r(n-1+\ell), \dots, r(\ell+1)]^T \in \mathbf{C}^n \\ &= [\mathbf{D}^{(\ell+1)T}(n-1) \quad r(\ell+1)]^T. \end{aligned} \quad (5.33)$$

5.2.2 Notation and Definitions for a Type I Configuration

In the Type I configuration, the notation is complicated by the fact that the adaptive filter input $u(n)$ is a function of the delay ℓ . The input data vector is

$$\mathbf{u}_M^\ell(i) = [u(i - \ell), u(i - 1 - \ell), \dots, u(i - M + 1 - \ell)]^T \quad (5.34)$$

and the errors are defined as

$$\begin{aligned} e_M(i, \ell) &= r(i) - \mathbf{w}_M^{\ell H}(n) \mathbf{u}_M^\ell(i) \\ \alpha_M(i, \ell) &= r(i) - \mathbf{w}_M^{\ell H}(n-1) \mathbf{u}_M^\ell(i). \end{aligned} \quad (5.35)$$

Note that, as in Chapter 3, each input sample in $\mathbf{u}_M^\ell(n)$ experiences the same delay ℓ . The data vector is not a function of ℓ and is

$$\mathbf{D}(n) = [r(n), r(n-1), \dots, r(1)]^T. \quad (5.36)$$

All the quantities defined in Subsection 5.2.1, and that are functions of $u(n)$, are now functions of ℓ . These quantities are $\mathbf{A}^\ell(n|i, M)$, $S^\ell(n|i, M)$, $\Phi_M^\ell(n)$, $F_m^\ell(n)$, $f_m^\ell(i)$, $\eta_m^\ell(i)$, $B_m^\ell(n)$, $b_m^\ell(i)$, $\psi_m^\ell(i)$, $\mathbf{E}_{M-1}^{f\ell}(n)$ and $\mathbf{E}_{M-1}^{b\ell}(n)$.

5.2.3 Geometrical Considerations

This subsection presents some definitions and considerations about projection operators in a Hilbert space. This *projection operator formalism* is used to derive geometrically the lag-recursive relations.

First, an inner product is defined in \mathbf{C}^n (\mathbf{C}^n exhibits an increasing dimensionality n). The inner product between two arbitrary vectors \mathbf{x} and \mathbf{y} is

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \mathbf{x}^H \mathbf{W}_n \mathbf{y} \\ &= \sum_{i=1}^n \beta^{n-i} x_i^* y_i, \end{aligned} \quad (5.37)$$

where the weighting matrix is

$$\mathbf{W}_n = \text{diag}[\beta^{n-1}, \beta^{n-2}, \dots, \beta^2, \beta, 1]. \quad (5.38)$$

Defining the norm of a vector \mathbf{x} as

$$\|\mathbf{x}\| = (\langle \mathbf{x}, \mathbf{x} \rangle)^{1/2}, \quad (5.39)$$

each n -dimensional vector in \mathbf{C}^n with finite components has a finite norm and \mathbf{C}^n is a Hilbert space [2] †. Denote the projection of a vector \mathbf{x} onto a subspace S as $\mathbf{P}_S \mathbf{x}$. The orthogonal projection of \mathbf{x} onto subspace S is written as

$$\mathbf{P}_S^\perp \mathbf{x} = \mathbf{x} - \mathbf{P}_S \mathbf{x}, \quad (5.40)$$

and is the error vector between \mathbf{x} and its projection on S . The projection of the vector \mathbf{y} on the vector \mathbf{x} is

$$\mathbf{P}_{\mathbf{x}} \mathbf{y} = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|^2} \mathbf{x}. \quad (5.41)$$

Two order updates for the projection operators are useful. They are based on the fact that the vector space spanned by a subspace S and a vector \mathbf{x} not in S , denoted $S \cup \{\mathbf{x}\}$, can be decomposed as [2]

$$S \cup \{\mathbf{x}\} = S \oplus \{\mathbf{P}_S^\perp \mathbf{x}\}, \quad (5.42)$$

where the notation $\{\mathbf{v}\}$ denotes the vector space spanned by \mathbf{v} . Since S and $\{\mathbf{P}_S^\perp \mathbf{x}\}$ are two orthogonal subspaces, the following order updates can be derived from geometrical considerations

$$\mathbf{P}_{S \cup \{\mathbf{x}\}} \mathbf{y} = \mathbf{P}_S \mathbf{y} + \mathbf{P}_{\{\mathbf{P}_S^\perp \mathbf{x}\}} \mathbf{y} \quad (5.43)$$

$$\mathbf{P}_{S \cup \{\mathbf{x}\}}^\perp \mathbf{y} = \mathbf{P}_S^\perp \mathbf{y} - \mathbf{P}_{\{\mathbf{P}_S^\perp \mathbf{x}\}} \mathbf{y}. \quad (5.44)$$

The linear least-square estimate of $\mathbf{D}^\ell(n)$, given the vectors $\mathbf{U}(n), z^{-1}\mathbf{U}(n), \dots, z^{-M+1}\mathbf{U}(n)$, is defined as the linear combination of those vectors which is closest to $\mathbf{D}^\ell(n)$ in the LS sense [2]. The optimum weight vector $\hat{\mathbf{w}}_M^\ell(n)$ is therefore the vector minimizing the norm of the error vector $\mathbf{E}_M^\ell(n)$, i.e., for a Type II-DRB adaptive system, the vector whose coefficients minimize

$$\mathcal{E}(n) = \|\mathbf{E}_M^\ell(n)\|^2 = \|\mathbf{D}^\ell(n) - \sum_{i=1}^M w_{iM}^{\ell*}(n) z^{-(i-1)} \mathbf{U}(n)\|^2. \quad (5.45)$$

The optimum LS estimate $\hat{\mathbf{D}}^\ell(n)$ is the projection on the subspace $S(n|0, M-1)$ of the vector $\mathbf{D}^\ell(n)$ [2]. Then, from (5.45), the following two projection equations emerge

$$\hat{\mathbf{D}}^\ell(n) = \sum_{i=1}^M \hat{w}_{iM}^{\ell*}(n) z^{-(i-1)} \mathbf{U}(n) = \mathbf{P}_{S(n|0, M-1)} \mathbf{D}^\ell(n) \quad (5.46)$$

$$\mathbf{E}_M^\ell(n) = \mathbf{P}_{S(n|0, M-1)}^\perp \mathbf{D}^\ell(n). \quad (5.47)$$

Note that

$$\begin{aligned} \hat{\mathbf{D}}^\ell(n) &= \sum_{i=1}^M \hat{w}_{iM}^{\ell*}(n) z^{-(i-1)} \mathbf{U}(n) \\ &= \mathbf{A}(n|0, M-1) \hat{\mathbf{w}}_M^{\ell*}(n). \end{aligned}$$

† Strictly speaking, a Hilbert space is an inner product space that is complete [64]. The vector space \mathbf{C}^n satisfies this condition, i.e. every Cauchy sequence of vectors converges in \mathbf{C}^n .

5.3 Geometrical Derivation of Lag-Recursive Relations

For a fixed block of data, it is possible to derive a series of recursions that compute the least sum of squared errors and the optimum LS weight vector at every possible lag, from the current values at lag ℓ . These recursions are derived, using vector and matrix manipulations, in [63].

Fast RLS adaptive filter algorithms can be derived using geometrical arguments. Cioffi and Kailath [61] derive the fast transversal filter using a geometrical method and Alexander [65] gives a tutorial review of the same subject. Another very good geometrical derivation is found in [2] and will be relied upon in this section. Lag-update relations are similarly derived in this section, for *on-line* computations of $\hat{\xi}_{M_o}(n, \ell + 1)$, $\hat{\xi}_{M_o}(n, \ell - 1)$, $\hat{\mathbf{w}}_M^{\ell+1}(n)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n)$ from $\hat{\xi}_{M_o}(n, \ell)$ and $\hat{\mathbf{w}}_M^\ell(n)$. In order to perform this new derivation, the projection operator formalism presented in Section 5.2.3 is used.

A first series of recursions, in term of the lag ℓ , is derived for the computation of $\hat{\xi}_{M_o}(n, \ell + 1)$ and $\hat{\xi}_{M_o}(n, \ell - 1)$, from $\hat{\xi}_{M_o}(n, \ell)$. A second series allows the computation of $\hat{\mathbf{w}}_M^{\ell+1}(n)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n)$, from $\hat{\mathbf{w}}_M^\ell(n)$. An alternate derivation is given in Appendix G and is based only on matrix manipulations.

The lag-recursive relations are first derived for a Type II-DRB system, because the derivation is simpler and gives results more readily applicable in practice. The lag-updates for a Type I configuration can be derived the same way. They are given and discussed in Subsection 5.3.2.

5.3.1 Derivation for a Type II-DRB Configuration

The derivation is first performed for the sum of squared errors. It is followed by a similar derivation for the LS weight vector.

5.3.1.1 Recursions for the Error

Using (5.30) and (5.44), (5.47) can be expressed as

$$\mathbf{E}_M^\ell(n) = \mathbf{P}_{\bar{S}(n|1, M-1)}^\perp \mathbf{D}^\ell(n) - \mathbf{P}_{\{\mathbf{P}_{\bar{S}(n|1, M-1)}^\perp \mathbf{U}(n)\}} \mathbf{D}^\ell(n). \quad (5.48)$$

Then, making use of (5.32) and (5.33),

$$\begin{aligned} \mathbf{P}_{\bar{S}(n|1, M-1)}^\perp \mathbf{D}^\ell(n) &= \mathbf{P}_{\bar{S}(n-1|0, M-2)}^\perp \mathbf{D}^{\ell+1}(n-1) \\ &= \mathbf{E}_{M-1}^{\ell+1}(n-1). \end{aligned} \quad (5.49)$$

Furthermore, the order $M - 1$ optimum LS one-step forward prediction of $u(n)$ is obtained through the projection of the vector $\mathbf{U}(n)$ on the subspace $S(n|1, M - 1)$ and the forward error prediction vector $\mathbf{E}_{M-1}^f(n)$ is given by

$$\mathbf{E}_{M-1}^f(n) = \mathbf{P}_{S(n|1, M-1)}^\perp \mathbf{U}(n). \quad (5.50)$$

Equation (5.48) can then be written as

$$\mathbf{E}_M^\ell(n) = \mathbf{E}_{M-1}^{\ell+1}(n-1) - \mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n). \quad (5.51)$$

Using (5.41), the following expression is obtained

$$\mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n) = \frac{\langle \mathbf{E}_{M-1}^f(n), \mathbf{D}^\ell(n) \rangle}{\|\mathbf{E}_{M-1}^f(n)\|^2} \mathbf{E}_{M-1}^f(n). \quad (5.52)$$

From the definition of the inner product (5.37), it is found that

$$\langle \mathbf{E}_{M-1}^f(n), \mathbf{D}^\ell(n) \rangle = \sum_{i=1}^n \beta^{n-i} f_{M-1}^*(i) r(i + \ell). \quad (5.53)$$

Define $v_{M-1}^{f\ell}(n)$ as the complex conjugate of the inner product of the forward error prediction vector and the desired response vector, i.e.

$$v_{M-1}^{f\ell*}(n) = \langle \mathbf{E}_{M-1}^f(n), \mathbf{D}^\ell(n) \rangle. \quad (5.54)$$

Also, referring to (5.22) and (5.28), it is seen that (using (5.37) and (5.39) for the norm definition)

$$\|\mathbf{E}_{M-1}^f(n)\|^2 = F_{M-1}(n). \quad (5.55)$$

Then, (5.52) can be written as

$$\mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n) = \frac{v_{M-1}^{f\ell*}(n)}{F_{M-1}(n)} \mathbf{E}_{M-1}^f(n). \quad (5.56)$$

Using (5.56) in (5.51) gives

$$\mathbf{E}_M^\ell(n) = \mathbf{E}_{M-1}^{\ell+1}(n-1) - \frac{v_{M-1}^{f\ell*}(n)}{F_{M-1}(n)} \mathbf{E}_{M-1}^f(n). \quad (5.57)$$

Noting that

$$\|\mathbf{E}_M^\ell(n)\|^2 = \hat{\xi}_{M_0}(n, \ell), \quad (5.58)$$

and taking the squared norm on both sides of (5.57), and because the vectors $\mathbf{E}_M^\ell(n)$ and $\mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n)$ are orthogonal (see (5.30), (5.48) and (5.51)),

$$\hat{\xi}_{M_0}(n, \ell) = \hat{\xi}_{(M-1)_0}(n-1, \ell+1) - \frac{|v_{M-1}^{f\ell}(n)|^2}{F_{M-1}(n)}, \quad (5.59)$$

which is the first recursion of interest. It gives $\hat{\xi}_{(M-1)o}(n-1, \ell+1)$ in terms of $\hat{\xi}_{M_o}(n, \ell)$.

A relation linking $\hat{\xi}_{(M-1)o}(n, \ell+1)$ to $\hat{\xi}_{M_o}(n, \ell+1)$ can be derived in a similar way. First, write (5.47) for $\ell+1$

$$\mathbf{E}_M^{\ell+1}(n) = \mathbf{P}_{S(n|0, M-1)}^\perp \mathbf{D}^{\ell+1}(n). \quad (5.60)$$

Then use (5.31) and (5.44) to write (5.60) as

$$\mathbf{E}_M^{\ell+1}(n) = \mathbf{P}_{S(n|0, M-2)}^\perp \mathbf{D}^{\ell+1}(n) - \mathbf{P}_{\{\mathbf{P}_{S(n|0, M-2)}^\perp z^{-M+1} \mathbf{U}(n)\}} \mathbf{D}^{\ell+1}(n). \quad (5.61)$$

Noting that

$$\mathbf{P}_{S(n|0, M-2)}^\perp \mathbf{D}^{\ell+1}(n) = \mathbf{E}_{M-1}^{\ell+1}(n) \quad (5.62)$$

and that

$$\mathbf{P}_{S(n|0, M-2)}^\perp z^{-M+1} \mathbf{U}(n) = \mathbf{E}_{M-1}^b(n), \quad (5.63)$$

then

$$\mathbf{E}_M^{\ell+1}(n) = \mathbf{E}_{M-1}^{\ell+1}(n) - \mathbf{P}_{\mathbf{E}_{M-1}^b(n)} \mathbf{D}^{\ell+1}(n). \quad (5.64)$$

Proceeding as in (5.52)

$$\mathbf{P}_{\mathbf{E}_{M-1}^b(n)} \mathbf{D}^{\ell+1}(n) = \frac{\langle \mathbf{E}_{M-1}^b(n), \mathbf{D}^{\ell+1}(n) \rangle}{\|\mathbf{E}_{M-1}^b(n)\|^2} \mathbf{E}_{M-1}^b(n), \quad (5.65)$$

which, defining $v_{M-1}^{b(\ell+1)*}(n)$ as

$$v_{M-1}^{b(\ell+1)*}(n) = \langle \mathbf{E}_{M-1}^b(n), \mathbf{D}^{\ell+1}(n) \rangle, \quad (5.66)$$

can be written as

$$\mathbf{P}_{\mathbf{E}_{M-1}^b(n)} \mathbf{D}^{\ell+1}(n) = \frac{v_{M-1}^{b(\ell+1)*}(n)}{B_{M-1}(n)} \mathbf{E}_{M-1}^b(n). \quad (5.67)$$

Then, (5.64) becomes

$$\mathbf{E}_M^{\ell+1}(n) = \mathbf{E}_{M-1}^{\ell+1}(n) - \frac{v_{M-1}^{b(\ell+1)*}(n)}{B_{M-1}(n)} \mathbf{E}_{M-1}^b(n), \quad (5.68)$$

and taking the squared norm on both sides of (5.68), and using the orthogonality of $\mathbf{E}_{M-1}^{\ell+1}(n)$ and $\mathbf{E}_{M-1}^b(n)$ gives

$$\hat{\xi}_{M_o}(n, \ell+1) = \hat{\xi}_{(M-1)o}(n, \ell+1) - \frac{|v_{M-1}^{b(\ell+1)*}(n)|^2}{B_{M-1}(n)}, \quad (5.69)$$

which is the third required recursion. It links $\hat{\xi}_{(M-1)o}(n, \ell+1)$ to $\hat{\xi}_{M_o}(n, \ell+1)$.

Pictorially, these derivations can be performed with the help of Figures 5.1 and 5.2. The subspaces $S(n|1, M-1)$ and $S(n|0, M-2)$ are represented as one-dimensional vector spaces. Then, the subspace $S(n|0, M-1)$ is the two-dimensional vector space spanned by $\mathbf{U}(n)$ and $S(n|1, M-1)$ or the one spanned by $z^{-M+1}\mathbf{U}(n)$ and $S(n|0, M-2)$. The vector $\mathbf{E}_{M-1}^f(n)$ is orthogonal to $S(n|1, M-1)$ and links the latter to $\mathbf{U}(n)$, while $\mathbf{E}_M^\ell(n)$ is orthogonal to $S(n|0, M-1)$ and joins $\mathbf{D}^\ell(n)$. The error vectors $\mathbf{E}_{M-1}^b(n)$ and $\mathbf{E}_{M-1}^{\ell+1}(n)$ are similarly represented in Figure 5.2. Then, the orthogonal equations (5.51) and (5.64) are obvious from the figures.

Finally, a time update recursion is necessary for $\hat{\xi}_{(M-1)o}(n-1, \ell+1)$. This recursion is common and can also be derived geometrically, although it requires more work than for the above recursions [2]. It is derived using matrix manipulations in Appendix G and involves both the *a priori* and *a posteriori* estimation errors. The recursion is

$$\hat{\xi}_{(M-1)o}(n, \ell+1) = \beta \hat{\xi}_{(M-1)o}(n-1, \ell+1) + \alpha_{M-1}^*(n, \ell+1) e_{M-1}(n, \ell+1). \quad (5.70)$$

Collecting (5.59), (5.70) and (5.69), the recursions for computing $\hat{\xi}_{M_o}(n, \ell+1)$ from $\hat{\xi}_{M_o}(n, \ell)$ are

$$\hat{\xi}_{(M-1)o}(n-1, \ell+1) = \hat{\xi}_{M_o}(n, \ell) + \frac{|v_{M-1}^{f\ell}(n)|^2}{F_{M-1}(n)} \quad (5.71)$$

$$\hat{\xi}_{(M-1)o}(n, \ell+1) = \beta \hat{\xi}_{(M-1)o}(n-1, \ell+1) + \alpha_{M-1}^*(n, \ell+1) e_{M-1}(n, \ell+1) \quad (5.72)$$

$$\hat{\xi}_{M_o}(n, \ell+1) = \hat{\xi}_{(M-1)o}(n, \ell+1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}(n)}. \quad (5.73)$$

Using the above expressions in reverse order gives the backward computation of the error.

$$\hat{\xi}_{(M-1)o}(n, \ell) = \hat{\xi}_{M_o}(n, \ell) + \frac{|v_{M-1}^{b\ell}(n)|^2}{B_{M-1}(n)} \quad (5.74)$$

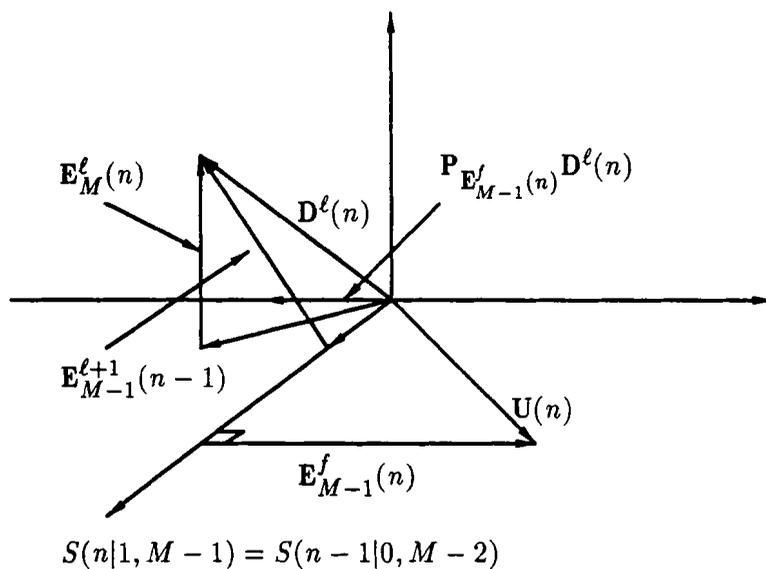
$$\hat{\xi}_{(M-1)o}(n-1, \ell) = \beta^{-1} \hat{\xi}_{(M-1)o}(n, \ell) - \beta^{-1} \alpha_{M-1}^*(n, \ell) e_{M-1}(n, \ell) \quad (5.75)$$

$$\hat{\xi}_{M_o}(n, \ell-1) = \hat{\xi}_{(M-1)o}(n-1, \ell) - \frac{|v_{M-1}^{f(\ell-1)}(n)|^2}{F_{M-1}(n)}. \quad (5.76)$$

5.3.1.2 Recursions for the LS Weight Vector

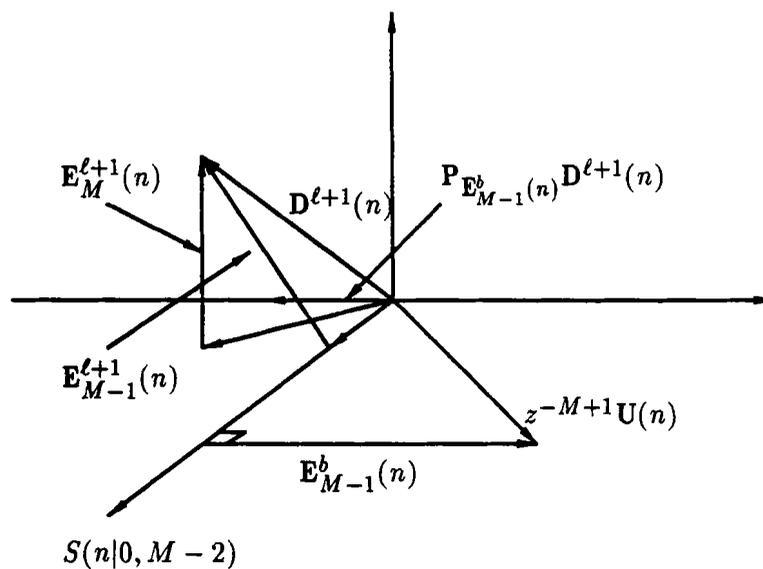
Figures 5.1 and 5.2 can also be used to perform the derivations of the weight vector recursions. From Figure 5.1, the following equation is obtained

$$\mathbf{A}(n|0, M-1) \hat{\mathbf{w}}_M^{\ell*}(n) = \mathbf{A}(n-1|0, M-2) \hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n-1) + \mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n), \quad (5.77)$$



$$\mathbf{E}_M^\ell(n) = \mathbf{E}_{M-1}^{\ell+1}(n-1) - \mathbf{P}_{\mathbf{E}_{M-1}^f(n)} \mathbf{D}^\ell(n)$$

Fig. 5.1 Geometrical interpretation of (5.51)



$$\mathbf{E}_M^{\ell+1}(n) = \mathbf{E}_{M-1}^{\ell+1}(n) - \mathbf{P}_{\mathbf{E}_{M-1}^b(n)} \mathbf{D}^{\ell+1}(n)$$

Fig. 5.2 Geometrical interpretation of (5.64)

where

$$\mathbf{A}(n|0, M-1)\hat{\mathbf{w}}_M^{\ell*}(n) = \mathbf{P}_{S(n|0, M-1)}\mathbf{D}^\ell(n) \quad (5.78)$$

and

$$\mathbf{A}(n-1|0, M-2)\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n-1) = \mathbf{P}_{S(n-1|0, M-2)}\mathbf{D}^\ell(n). \quad (5.79)$$

Using (5.52) in (5.77) and noting that

$$\mathbf{E}_{M-1}^f(n) = \mathbf{U}(n) - \mathbf{A}(n-1|0, M-2)\mathbf{a}_{M-1}^*(n), \quad (5.80)$$

the following expression is obtained

$$\begin{aligned} & \mathbf{A}(n|0, M-1)\hat{\mathbf{w}}_M^{\ell*}(n) = \\ & \mathbf{A}(n-1|0, M-2)\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n-1) + \frac{\langle \mathbf{E}_{M-1}^f(n), \mathbf{D}^\ell(n) \rangle}{F_{M-1}(n)} [\mathbf{U}(n) - \mathbf{A}(n-1|0, M-2)\mathbf{a}_{M-1}^*(n)]. \end{aligned} \quad (5.81)$$

This equation can also be written as

$$\begin{aligned} & \hat{\mathbf{w}}_{1M}^{\ell*}(n)\mathbf{U}(n) + \mathbf{A}(n-1|0, M-2)[\hat{\mathbf{w}}_M^{\ell*}(n)]_{M-1} = \\ & \mathbf{A}(n-1|0, M-2) \left[\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n-1) - \mathbf{a}_{M-1}^*(n) \frac{v_{M-1}^{f\ell*}(n)}{F_{M-1}(n)} \right] + \frac{v_{M-1}^{f\ell*}(n)}{F_{M-1}(n)} \mathbf{U}(n), \end{aligned} \quad (5.82)$$

where $[\mathbf{v}]_{M-1}$ stands for the vector made of the $M-1$ last components of the vector \mathbf{v} .

Equating similar terms, the following recursion is obtained

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}(n)\hat{\mathbf{w}}_{1M}^\ell(n), \quad (5.83)$$

along with

$$\hat{\mathbf{w}}_{1M}^\ell(n) = \frac{v_{M-1}^{f\ell}(n)}{F_{M-1}(n)}. \quad (5.84)$$

Equation (5.83) is the recursion linking $\hat{\mathbf{w}}_M^\ell(n)$ to $\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1)$.

Similarly, from Figure 5.2, the following is obtained

$$\begin{aligned} \mathbf{A}(n|0, M-1)\hat{\mathbf{w}}_M^{(\ell+1)*}(n) &= \mathbf{A}(n|0, M-2)\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n) + \mathbf{P}_{\mathbf{E}_{M-1}^b(n)}\mathbf{D}^{\ell+1}(n) \\ &= \mathbf{A}(n|0, M-2)\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n) + \frac{\langle \mathbf{E}_{M-1}^b(n), \mathbf{D}^{\ell+1}(n) \rangle}{B_{M-1}(n)} \mathbf{E}_{M-1}^b(n) \\ &= \mathbf{A}(n|0, M-2)\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n) \\ & \quad + \frac{v_{M-1}^{b(\ell+1)*}(n)}{B_{M-1}(n)} [z^{-M+1}\mathbf{U}(n) - \mathbf{A}(n|0, M-2)\mathbf{b}_{M-1}^*(n)]. \end{aligned} \quad (5.85)$$

Write (5.85) as

$$\begin{aligned} & \mathbf{A}(n|0, M-2) [\hat{\mathbf{w}}_M^{(\ell+1)*}(n)]_{M-1} + z^{-M+1} \mathbf{U}(n) \hat{w}_{MM}^{(\ell+1)*}(n) = \\ & \mathbf{A}(n|0, M-2) \left[\hat{\mathbf{w}}_{M-1}^{(\ell+1)*}(n) - \frac{v_{M-1}^{b(\ell+1)*}(n)}{B_{M-1}(n)} \mathbf{b}_{M-1}^*(n) \right] + \frac{v_{M-1}^{b(\ell+1)*}(n)}{B_{M-1}(n)} z^{-M+1} \mathbf{U}(n), \end{aligned} \quad (5.86)$$

where $[\hat{\mathbf{w}}_M^\ell(n)]_{M-1}$ is defined as the $(M-1)$ -vector corresponding to the *first* components of $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{w}_{MM}^\ell(n)$ is the M^{th} component of the same vector. Equating similar terms, the following equations are obtained

$$\hat{w}_{MM}^{\ell+1}(n) = \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}(n)} \quad (5.87)$$

$$[\hat{\mathbf{w}}_M^{\ell+1}(n)]_{M-1} = \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) - \hat{w}_{MM}^{\ell+1}(n) \mathbf{b}_{M-1}(n). \quad (5.88)$$

Then, by combining these two equations, a recursion linking $\hat{\mathbf{w}}_{M-1}^{\ell+1}(n)$ and $\hat{\mathbf{w}}_M^{\ell+1}(n)$ is obtained. It is

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix}. \quad (5.89)$$

The recursion necessary to link (5.83) and (5.89) is a common time update recursion and involves the Kalman gain vector $\mathbf{g}_{M-1}(n)$ and the *a posteriori* estimation error $e_{M-1}(n, \ell+1)$ and is [7]

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) = \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \beta^{-1} \mathbf{g}_{M-1}(n) e_{M-1}(n, \ell+1). \quad (5.90)$$

Collecting (5.83), (5.90) and (5.89), the set of recursions for the upward weight vector computation is

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}(n) \hat{w}_{1M}^\ell(n) \quad (5.91)$$

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) = \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \beta^{-1} \mathbf{g}_{M-1}(n) e_{M-1}(n, \ell+1) \quad (5.92)$$

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} \quad (5.93).$$

Using the upward recursions in reverse order, the following two downward recursions are obtained

$$\hat{\mathbf{w}}_{M-1}^\ell(n) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{b}_{M-1}(n) \hat{w}_{MM}^\ell(n) \quad (5.94)$$

$$\hat{\mathbf{w}}_M^{\ell-1}(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^\ell(n-1) \end{bmatrix} + \frac{v_{M-1}^{f(\ell-1)}(n)}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix}. \quad (5.95)$$

5.3.2 Lag Recursions for a Type I Configuration

Following a procedure similar to the previous one, the following set of lag-update recursions for the error and for the LS weight vector can be obtained.

5.3.2.1 Recursions for the Error

$$\hat{\xi}_{(M-1)o}(n, \ell + 1) = \hat{\xi}_{Mo}(n, \ell) + \frac{|v_{M-1}^{f\ell}(n)|^2}{F_{M-1}^\ell} \quad (5.96)$$

$$\hat{\xi}_{Mo}(n, \ell + 1) = \hat{\xi}_{(M-1)o}(n, \ell + 1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}^{\ell+1}(n)} \quad (5.97)$$

Using the above expressions in reverse order gives the backward computation of the error.

$$\hat{\xi}_{(M-1)o}(n, \ell) = \hat{\xi}_{Mo}(n, \ell) + \frac{|v_{M-1}^{b\ell}(n)|^2}{B_{M-1}^\ell} \quad (5.98)$$

$$\hat{\xi}_{Mo}(n, \ell - 1) = \hat{\xi}_{(M-1)o}(n, \ell) - \frac{|v_{M-1}^{f(\ell-1)}(n)|^2}{F_{M-1}^{\ell-1}}. \quad (5.99)$$

5.3.2.2 Recursions for the LS Weight Vector

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}^\ell(n) \hat{w}_{1M}^\ell(n) \quad (5.100)$$

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}^{\ell+1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}^{\ell+1}(n) \\ 1 \end{bmatrix} \quad (5.101)$$

Using the upward recursions in reverse order, the following two recursions are obtained

$$\hat{\mathbf{w}}_{M-1}^\ell(n) = [\hat{\mathbf{w}}_M^{\ell+1}(n)]_{M-1} + \mathbf{b}_{M-1}^\ell(n) \hat{w}_{MM}^{\ell+1}(n) \quad (5.102)$$

$$\hat{\mathbf{w}}_M^{\ell-1}(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^\ell(n) \end{bmatrix} + \frac{v_{M-1}^{f(\ell-1)}(n)}{F_{M-1}^{\ell-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}^{\ell-1}(n) \end{bmatrix}. \quad (5.103)$$

Note that the main difference in the lag-update relations between the two types of systems lies in the fact that no time-update equations as (5.72), (5.75) or (5.92) is required in the Type I relationships.

5.3.3 Discussion

Recall that the on-line lag-update recursions can also be derived using matrix manipulations, as it is performed in Appendix G. It is interesting to relate the properties of each of the two approaches. For the matrix manipulations derivation of Appendix G, the key equation is the shift invariance (E.4) given in Appendix E in which a lag ℓ cross-correlation vector is partitioned in terms of a $\ell + 1$ cross-correlation vector. In the geometrical approach, the key equation is (5.33) and relates a lag ℓ desired response vector to a lag $\ell + 1$ desired response vector. In both cases, the lag $\ell + 1$ vector is given for time $n - 1$ and involves $(M - 1)$ -order prediction (see (E.4) and (5.49)). Therefore, time update and order update relations are necessary steps in the lag update, for a Type II-DRB adaptive system configuration. In the case of a Type I system, the key relations are

$$\boldsymbol{\theta}_M^\ell(n) = \begin{bmatrix} \bar{\theta}_M^\ell(n) \\ \boldsymbol{\theta}_{M-1}^{\ell+1}(n) \end{bmatrix}, \quad (5.104)$$

since

$$\mathbf{u}_M^\ell(i) = \begin{bmatrix} u(i - \ell) \\ \mathbf{u}_{M-1}^{\ell+1}(i) \end{bmatrix}, \quad (5.105)$$

and

$$S^\ell(n|1, M - 1) = S^{\ell+1}(n|0, M - 2), \quad (5.106)$$

since

$$\mathbf{U}^\ell(n - 1) = \mathbf{U}^{\ell+1}(n). \quad (5.107)$$

These relations show that, in the Type I case, $M - 1$ -order predictors are still required, but that the time $n - 1$ is not involved anymore.

The geometrical derivations give a picture of how the $(M - 1)$ -order predictors get involved in the algorithm. Considering Figures 5.1 and 5.2, if an initial relation starting with $\mathbf{E}_M^\ell(n)$ (or $\hat{\xi}_{M_0}(n, \ell)$) is required, it is natural to express it as a function of $\mathbf{E}_{M-1}^{\ell+1}(n)$ and $\mathbf{E}_{M-1}^f(n)$. Similarly, it is natural to express the required vector $\mathbf{E}_M^{\ell+1}(n)$ in terms of $\mathbf{E}_{M-1}^{\ell+1}(n)$ and $\mathbf{E}_{M-1}^b(n)$. This gives a relation involving the error for the current lag ℓ and another involving the error for the updated lag $\ell + 1$. The relation linking these two equations nicely involves the time update of $\mathbf{E}_{M-1}^{\ell+1}(n - 1)$ in the Type II-DRB case and no time update in the Type I case. Such nice and simple interrelations between the variables of the algorithm do not seem to exist for M -order predictors.

Note that the lag-recursive relations, for both the errors and the weight vectors, mostly involve parameters and quantities that are computed by the FTF algorithm (see Appendix F). One major difference resides in the order of the predictors, which is $M - 1$

in the lag-recursive equations. But the FTF can be redefined easily for $(M - 1)$ -order predictors, as indicated in the next section.

5.4 Joint Time Delay Estimation and Adaptive RLS Algorithms with the Lag-Recursive Relations in Type II-DRB Configuration

Based on the error and weight vector recursions developed in the previous section, different variants of joint time delay and FTF algorithms can be obtained. These algorithms are composed of three distinct computational phases. The first phase is essentially the preliminary computations phase of the FTF algorithm, given in equation (F.1) of Appendix F for M -order predictors. In the joint algorithm, this order is changed to $M - 1$. The second computational phase involves the computation of the current weight vector $\hat{\mathbf{w}}_M^\ell(n)$ and the computation of the three errors $\hat{\xi}_{M_o}(n, \ell)$, $\hat{\xi}_{M_o}(n, \ell + 1)$ and $\hat{\xi}_{M_o}(n, \ell - 1)$. These computations are performed by using the lag update recursions for the error and the weight vector. In the joint algorithms considered in this chapter, the computation of $\hat{\mathbf{w}}_M^{\ell-1}(n)$ and $\hat{\xi}_{M_o}(n, \ell - 1)$ is first performed, using the usual FTF equations. Then the upward lag recursions for both the error and the weight vector are used twice, in order to get the errors for ℓ and $\ell + 1$ and the weight vector for ℓ . These successive applications of the upward recursions produce the least number of computations, compared for example to the application of the upward and downward recursions on the error and weight vector at lag ℓ . This choice also simplifies the third computational phase, which involves a decision on the lag update and the computations of the new corresponding variables.

The joint algorithm is given only for a Type II-DRB configuration, since the corresponding algorithm for a Type I system can be expressed in a straightforward manner. Note however that when the lag gets updated in the latter system, the variables involved in the preliminary computations phase have to be updated also. This produces a serious increase in the computational complexity and makes the joint Type I system not very appealing in practice.

Schematically, the preliminary and error computations phases of the algorithm can be represented as in Figure 5.3, where six parallel digital filter are represented. The top three filters are essentially the same as the ones used in the conventional fast transversal filter [61], [7], except for the difference in predictors order (compare Figures 5.3 and F.1). The fourth filter is for the computation of $\hat{\xi}_{M_o}(n, \ell - 1)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n - 1)$. Notice that $\hat{\xi}_{(M-1)_o}(n - 1, \ell)$ is also obtained from that filter, using (5.71) and (5.84). A fifth filter, with weight vector $\hat{\mathbf{w}}_{M-1}^\ell(n - 1)$ obtained from (5.91), is used to obtain $v_{M-1}^{b\ell}(n)$, from which $\hat{\xi}_{M_o}(n, \ell)$, $\hat{\mathbf{w}}_M^\ell(n)$

and $\hat{\xi}_{(M-1)o}(n, \ell + 1)$ are computed. Finally a sixth transversal filter, with weight vector $\hat{w}_{M-1}^{\ell+1}(n-1)$, is used in the computation of $v_{M-1}^{b(\ell+1)}(n)$ and $\hat{\xi}_{Mo}(n, \ell + 1)$.

The joint algorithm, based on Fig. 5.3, is given in the next subsection. Parts a) and b) of this algorithm correspond to the figure, while part c) constitutes the lag update section. The decision about this update may involve the time average of the sum of squared errors, as indicated in Section 5.4.1, or another form of average. Note that in the case of positive update, in (5.113), only a simple transfer of information from $\ell + 1$ quantities to ℓ ones and the reinitialization of certain variables, are required. In the case of negative update, in (5.114), some intermediate computations, involving $\theta_M^{\ell-1}(n)$ and $\theta_M^{\ell-2}(n)$, are necessary. These quantities are used with some of the backward lag-recursive relations, in the computation of the new values of $w_M^{\ell-1}(n)$ and $\hat{\xi}_{Mo}(n, \ell - 1)$.

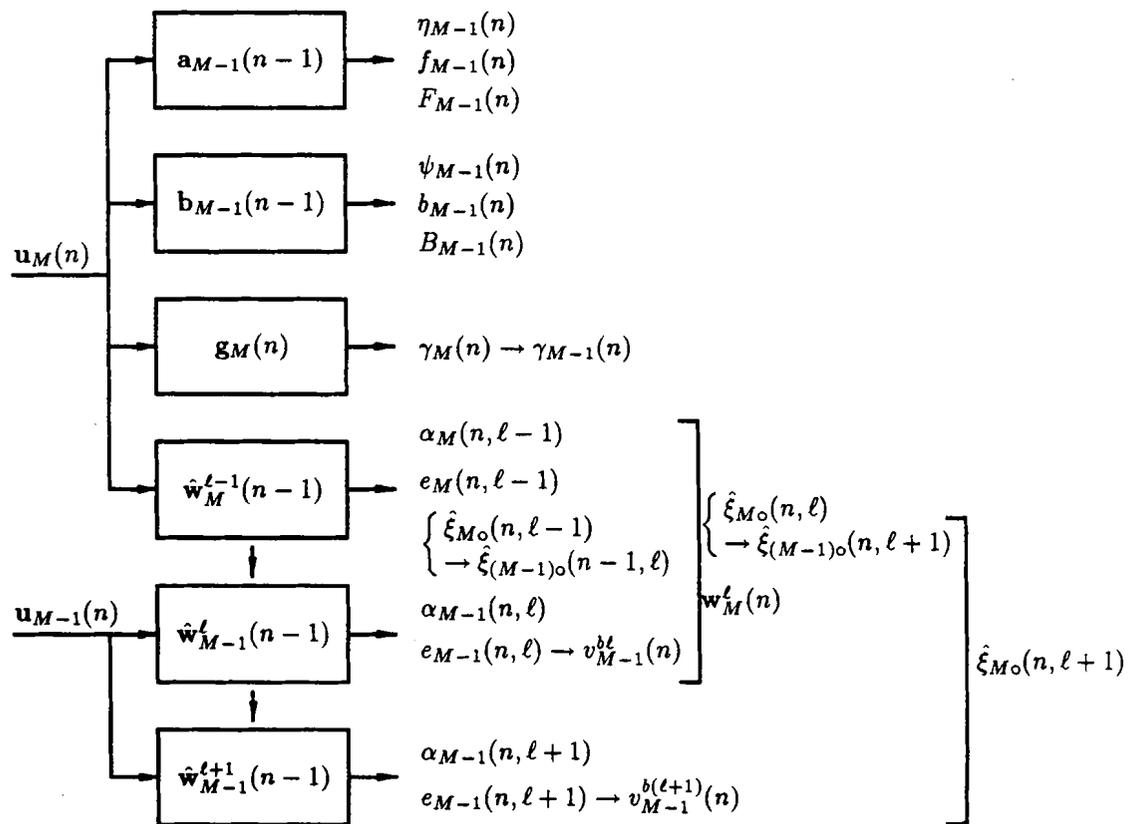


Fig. 5.3 Interpretation of the lag $\ell - 1$, ℓ and $\ell + 1$ error computations, in terms of transversal filters

5.4.1 The Joint Algorithm for a Type II-DRB Configuration

a) Preliminary Computations

$$\begin{aligned}
\eta_{M-1}(n) &= u(n) - \mathbf{a}_{M-1}^H(n-1)\mathbf{u}_{M-1}(n-1) \\
f_{M-1}(n) &= \frac{\eta_{M-1}(n)}{\gamma_{M-1}(n-1)} \\
\mathbf{a}_{M-1}(n) &= \mathbf{a}_{M-1}(n-1) + \beta^{-1}\mathbf{g}_{M-1}(n-1)f_{M-1}^*(n) \\
F_{M-1}(n) &= \beta F_{M-1}(n-1) + \eta_{M-1}(n)f_{M-1}^*(n) \\
\mathbf{g}_M(n) &= \begin{bmatrix} 0 \\ \mathbf{g}_{M-1}(n-1) \end{bmatrix} + \frac{\eta_{M-1}(n)}{F_{M-1}(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n-1) \end{bmatrix} \\
\mathbf{g}_{M-1}(n) &= [\mathbf{g}_M(n)]_{M-1} + g_{MM}(n)\mathbf{b}_{M-1}(n-1) \\
\gamma_M(n) &= \gamma_{M-1}(n-1) + \frac{|\eta_{M-1}(n)|^2}{\beta F_{M-1}(n-1)} \\
\psi_{M-1}(n) &= g_{MM}(n)B_{M-1}(n-1) \\
\gamma_{M-1}(n) &= \gamma_M(n) - \beta^{-1}g_{MM}(n)\psi_{M-1}^*(n) \\
b_{M-1}(n) &= \frac{\psi_{M-1}(n)}{\gamma_{M-1}(n)} \\
\mathbf{b}_{M-1}(n) &= \mathbf{b}_{M-1}(n-1) + \beta^{-1}\mathbf{g}_{M-1}(n)b_{M-1}^*(n) \\
B_{M-1}(n) &= \beta B_{M-1}(n-1) + \psi_{M-1}(n)b_{M-1}^*(n)
\end{aligned} \tag{5.108}$$

b) Errors and weight vector computations

Extra recursions for update smoothness

$$\begin{aligned}
\theta_M^{\ell-1}(n) &= \beta\theta_M^{\ell-1}(n-1) + \mathbf{u}_M(n)r^*(n+\ell-1) \\
\theta_M^{\ell-2}(n) &= \beta\theta_M^{\ell-2}(n-1) + \mathbf{u}_M(n)r^*(n+\ell-2)
\end{aligned} \tag{5.109}$$

Lag $\ell - 1$ computations

$$\begin{aligned}
\alpha_M(n, \ell-1) &= r(n+\ell-1) - \hat{\mathbf{w}}_M^{(\ell-1)H}(n-1)\mathbf{u}_M(n) \\
e_M(n, \ell-1) &= \frac{\alpha_M(n, \ell-1)}{\gamma_M(n)} \\
\hat{\mathbf{w}}_M^{\ell-1}(n) &= \hat{\mathbf{w}}_M^{\ell-1}(n-1) + \beta^{-1}\mathbf{g}_M(n)e_M^*(n, \ell-1) \\
\hat{\xi}_{M_0}(n, \ell-1) &= \beta\hat{\xi}_{M_0}(n-1, \ell-1) + \alpha_M^*(n, \ell-1)e_M(n, \ell-1)
\end{aligned} \tag{5.110}$$

Lag ℓ computations

$$\begin{aligned}
\hat{\mathbf{w}}_{M-1}^\ell(n-1) &= [\hat{\mathbf{w}}_{M-1}^{\ell-1}(n)]_{M-1} + \mathbf{a}_{M-1}(n)w_{1M}^{\ell-1}(n) \\
\alpha_{M-1}(n, \ell) &= r(n+\ell) - \hat{\mathbf{w}}_{M-1}^{\ell H}(n-1)\mathbf{u}_{M-1}(n) \\
e_{M-1}(n, \ell) &= \frac{\alpha_{M-1}(n, \ell)}{\gamma_{M-1}(n)} \\
\hat{\mathbf{w}}_{M-1}^\ell(n) &= \hat{\mathbf{w}}_{M-1}^\ell(n-1) + \beta^{-1}\mathbf{g}_{M-1}(n)e_{M-1}^*(n, \ell) \\
v_{M-1}^{b\ell}(n) &= \beta v_{M-1}^{b\ell}(n-1) + \psi_{M-1}(n)e_{M-1}^*(n, \ell) \\
\hat{\mathbf{w}}_M^\ell(n) &= \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^\ell(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b\ell}(n)}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} \\
\hat{\xi}_{(M-1)_o}(n-1, \ell) &= \hat{\xi}_{M_o}(n, \ell-1) + F_{M-1}(n)|w_{1M}^{\ell-1}(n)|^2 \\
\hat{\xi}_{(M-1)_o}(n, \ell) &= \beta \hat{\xi}_{(M-1)_o}(n-1, \ell) + \gamma_{M-1}(n)|e_{M-1}(n, \ell)|^2 \\
\hat{\xi}_{M_o}(n, \ell) &= \hat{\xi}_{(M-1)_o}(n, \ell) - \frac{|v_{M-1}^{b\ell}(n)|^2}{B_{M-1}(n)}
\end{aligned} \tag{5.111}$$

Lag $\ell + 1$ computations

$$\begin{aligned}
\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) &= [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}(n)w_{1M}^\ell(n) \\
\alpha_{M-1}(n, \ell+1) &= r(n+\ell+1) - \hat{\mathbf{w}}_{M-1}^{(\ell+1)H}(n-1)\mathbf{u}_{M-1}(n) \\
e_{M-1}(n, \ell+1) &= \frac{\alpha_{M-1}(n, \ell+1)}{\gamma_{M-1}(n)} \\
\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) &= \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) + \beta^{-1}\mathbf{g}_{M-1}(n)e_{M-1}^*(n, \ell+1) \\
v_{M-1}^{b(\ell+1)}(n) &= \beta v_{M-1}^{b(\ell+1)}(n-1) + \psi_{M-1}(n)e_{M-1}^*(n, \ell+1) \\
\hat{\xi}_{(M-1)_o}(n-1, \ell+1) &= \hat{\xi}_{M_o}(n, \ell) + F_{M-1}(n)|w_{1M}^\ell(n)|^2 \\
\hat{\xi}_{(M-1)_o}(n, \ell+1) &= \beta \hat{\xi}_{(M-1)_o}(n-1, \ell+1) + \gamma_{M-1}(n)|e_{M-1}(n, \ell+1)|^2 \\
\hat{\xi}_{M_o}(n, \ell+1) &= \hat{\xi}_{(M-1)_o}(n, \ell+1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}(n)}
\end{aligned} \tag{5.112}$$

c) Updates

If $\langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle$ and $\langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle$ then

$$\begin{aligned}
 \hat{\mathbf{w}}_M^{\ell-1}(n) &\leftarrow \hat{\mathbf{w}}_M^\ell(n) \\
 \hat{\xi}_{M_o}(n, \ell - 1) &\leftarrow \hat{\xi}_{M_o}(n, \ell) \\
 v_{M-1}^{b\ell}(n) &\leftarrow v_{M-1}^{b(\ell+1)}(n) \\
 \theta_M^{\ell-2}(n) &\leftarrow \theta_M^{\ell-1}(n) \\
 \theta_M^{\ell-1}(n) &= \mathbf{u}_M(n)r^*(n + \ell) \\
 v_{M-1}^{b(\ell+1)}(n) &= 0 \\
 \ell &\leftarrow \ell + 1
 \end{aligned} \tag{5.113}$$

Endif

If $\langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle$ and $\langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle$ then

$$\begin{aligned}
 v_{M-1}^{b(\ell+1)}(n) &\leftarrow v_{M-1}^{b\ell}(n) \\
 v_{M-1}^{b\ell}(n) &= [-\mathbf{b}_{M-1}^H(n) \quad 1]\theta_M^{\ell-1}(n) \\
 v_{M-1}^{f(\ell-2)}(n) &= [1 \quad -\mathbf{a}_{M-1}^H(n)]\theta_M^{\ell-2}(n) \\
 \hat{\xi}_{(M-1)_o}(n, \ell - 1) &= \hat{\xi}_{M_o}(n, \ell - 1) + B_{M-1}(n)|w_{MM}^{\ell-1}(n)|^2 \\
 \hat{\mathbf{w}}_{M-1}^{\ell-1}(n - 1) &= [\hat{\mathbf{w}}_M^{\ell-1}(n - 1)]_{M-1} + \mathbf{b}_{M-1}(n - 1)w_{MM}^{\ell-1}(n - 1) \\
 \hat{\mathbf{w}}_M^{\ell-1}(n) &= \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^{\ell-1}(n - 1) \end{bmatrix} + \frac{v_{M-1}^{f(\ell-2)}(n)}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix} \\
 \hat{\xi}_{(M-1)_o}(n - 1, \ell - 1) &= \beta^{-1}\hat{\xi}_{(M-1)_o}(n, \ell - 1) - \beta^{-1} \frac{|\mathbf{u}_{M-1}^H(n)\hat{\mathbf{w}}_{M-1}^{\ell-1}(n - 1) - r^*(n + \ell - 1)|^2}{\gamma_{M-1}(n)} \\
 \hat{\xi}_{M_o}(n, \ell - 1) &= \hat{\xi}_{(M-1)_o}(n - 1, \ell - 1) - \frac{|v_{M-1}^{f(\ell-2)}(n)|^2}{F_{M-1}(n)} \\
 \theta_M^{\ell-1}(n) &\leftarrow \theta_M^{\ell-2}(n) \\
 \theta_M^{\ell-2}(n) &= \mathbf{u}_M(n)r^*(n + \ell - 3) \\
 \ell &\leftarrow \ell - 1
 \end{aligned} \tag{5.114}$$

Endif ■

5.4.2 Discussion

The originality of the joint LS algorithm presented in Subsection 5.4.1 resides in the serial computations, from $\hat{\mathbf{w}}_M^{\ell-1}(n - 1)$, of all the necessary errors and weight vectors for lags

ℓ and $\ell + 1$. One consequence of this serial approach is a reduction in the memory needed to store the different quantities of interest. The lag-update recursions append themselves nicely to the FTF algorithm of the form given in Appendix F. Note however that two extra recursions (equations (5.109)) are necessary to ensure update smoothness when the lag is updated from ℓ to $\ell - 1$ (equations (5.114)). In this case, the quantities $\theta_M^{\ell-1}(n)$ and $\theta_M^{\ell-2}(n)$ are necessary to update $v_{M-1}^{b\ell}(n)$ and to compute $v_{M-1}^{f(\ell-2)}(n)$ (necessary to update $\hat{w}_M^{\ell-1}(n)$). Note also that $\theta_M^{\ell-1}(n)$, $\theta_M^{\ell-2}(n)$ and $v_{M-1}^{b(\ell+1)}(n)$ must be reinitialized in the case of lag update (in equations (5.113) and (5.114)). These reinitializations constitute the only approximations of the joint LS algorithm and are justified by the limited memory of the algorithm (defined by β). Furthermore, the reinitialization of the cross-correlation vectors does not involve any of the algorithm's internal variables since the input signal $u(n)$ and the reference signal $r(n)$ are the only variables used in these computations.

In contrast, the application of three parallel versions of the RLS algorithm, one for each possible lag, requires the initialization of both the sum of squared errors and the weight vector, when the lag is updated. The initialization must be done assuming zero input data. This typically introduces an error in both of these quantities because their computation involves the internal variables $\gamma_M(n)$ and $\mathbf{g}_M(n)$ (see equations (5.110)), that were obtained from a totally different set of initial conditions (non-zero input data). In order to allow a smooth transition in the case of lag update, two extra parallel branches, one for $\ell + 2$ and one for $\ell - 2$, must be computed, which gives a final parallel algorithm involving five branches. This algorithm requires a fair amount of memory in order to store all the previous values of the variables used in the errors and weight vectors computation (equation (5.110)).

At the start of the joint algorithm, the internal variables of the FTF are initialized *exactly* as proposed by Cioffi [61], and the extra error and correlation variables are initialized to zero.

Finally, it is a custom with fast RLS algorithms to establish their computational complexity and to compare it to other types of algorithms. The complexity of the joint LS algorithm can be compared here to the that of the simple FTF algorithm. As in [7], this complexity is measured by the number of operations required to perform one iteration of the algorithm. An operation is either a multiplication, a division or an addition/subtraction. It is further assumed that all signals are real-valued. The operation count of the joint RLS algorithm of Subsection 5.4.1 is presented in Table 5.1, along with the counts for the simple FTF algorithm and for the parallel application of five RLS algorithms, in FTF form and in LS lattice form. These figures concern only the first two phases of the algorithms, i.e. the

preliminary and the errors and weight vectors computations phases. This choice reflects the fact that in tracking mode, the lag update is expected to be performed after many iterations, and therefore does not increase the computational count significantly.

Algorithm	Number of operations per iteration		
	Multiplications	Divisions	Additions/Subtractions
Simple FTF	$7M+6$	9	$6M+3$
Joint LS (5.4.1)	$16M+17$	16	$16M+2$
Parallel FTF	$15M+14$	17	$15M+5$
Parallel Lattice	$22M-7$	$11M-6$	$16M-6$

Table 5.1 Comparison between the computational complexities of the ordinary FTF algorithm, the joint time delay and FTF RLS algorithm of Section 5.4.1 and the parallel FTF and Lattice algorithms.

This table shows that the joint algorithm is twice as computationally involved as the FTF algorithm of Appendix F (with $(M - 1)$ -order predictors). It also shows that the parallel FTF algorithm and the joint LS algorithm are about as computationally intensive and that the lattice-based parallel algorithm is much more computationally involved.

5.5 Analysis of the Joint LS Algorithm in Steady-State

The convergence of the two estimates produced by the joint LS algorithm is studied in this section. In so doing, Assumptions 1 to 7 of Section 4.3 are retained, with the reference delay D being equal to an integer number of sampling periods.

5.5.1 The Joint LS Algorithm in Type II-DRB Configuration

The algorithm is studied in two phases; the adaptive delay estimate is considered first, followed by the adaptive filter analysis. The results are then used to obtain the excess MSE produced by the joint algorithm. The next section does not give a full analysis of

the LS delay estimation, but it points out the factors that influence the estimate mean and variance. The adaptive filter analysis, in Subsection 5.5.1.2, is more complete.

5.5.1.1 Considerations about the LS Delay Estimator in Steady-State

Considering the joint algorithm of Section 5.4, the delay estimate is obtained by comparing the three random variables $\langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle$, $\langle \hat{\xi}_{M_o}(n, \ell) \rangle$ and $\langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle$. A typical form of the function $\langle \hat{\xi}_{M_o}(n, d_n) \rangle$ is illustrated in Fig.5.4. It has a minimum equal to $\langle \hat{\xi}_{M_o}(n, D) \rangle$ and was obtained with the system parameters described in Section 6.2.

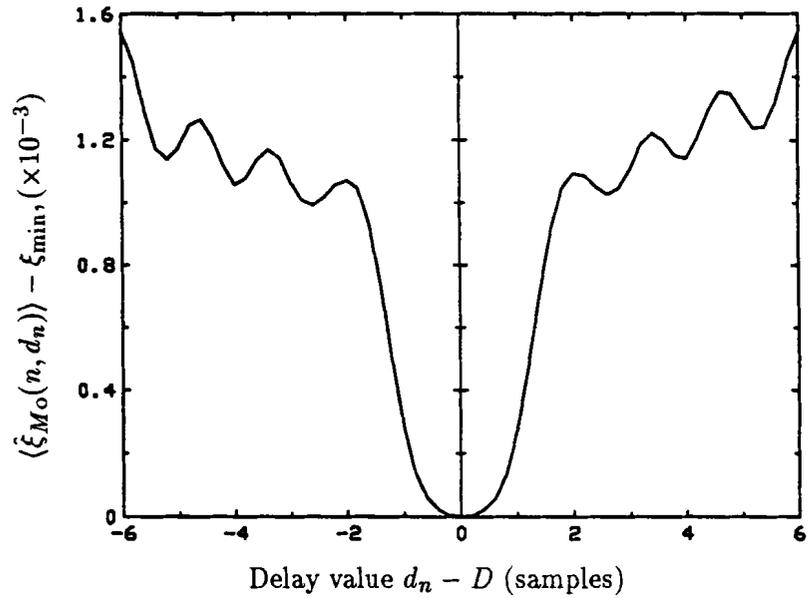


Fig. 5.4 Minimum sum of squared errors versus the continuous delay d_n , $\beta = 0.9$

Assuming that the adaptive delay is initially equal to the value ℓ , the probability of staying at this value is given by

$$P_{\mathcal{U}}(n) = \Pr\{\{\langle \hat{\xi}_{M_o}(n, \ell) \rangle < \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle\} \cap \{\langle \hat{\xi}_{M_o}(n, \ell) \rangle < \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle\}\} \quad (5.115)$$

and the probability of going from ℓ to $\ell + 1$ or $\ell - 1$ is given respectively by

$$P_{\ell(\ell+1)}(n) = \Pr\{\{\langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle\} \cap \{\langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle\}\} \quad (5.116)$$

$$P_{\ell(\ell-1)}(n) = \Pr\{\{\langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell) \rangle\} \cap \{\langle \hat{\xi}_{M_o}(n, \ell - 1) \rangle < \langle \hat{\xi}_{M_o}(n, \ell + 1) \rangle\}\}. \quad (5.117)$$

Because the variables $\hat{\xi}_{M_o}(n, \ell)$ are obtained from a first order difference equation (equation of the form of (5.70)), the transitions from one delay value to the other can be represented as a Markov chain [66]. The corresponding state-diagram has a state for each possible delay value and the transition probabilities are computed as in (5.115) to (5.117). The transition probability matrix is a band matrix, with nonzero entries on the main diagonal and on the two adjacent diagonals. The transition probabilities are functions of the input signal and noise statistics. Assuming, as in Chapter 4, that the MSE function has a symmetrical global minimum at $d_n = D$, and that there is no occurrence of false lock on any local minimum, then the delay estimator is unbiased and its variance is a function of the steady-state probabilities of being in the different states.

5.5.1.2 Analysis for the LS Adaptive Filter in Steady-State

From equation (F.2), the weight vector is updated as

$$\hat{\mathbf{w}}_M^\ell(n) = \hat{\mathbf{w}}_M^\ell(n-1) + \beta^{-1} \mathbf{g}_M(n) e_M^*(n, \ell), \quad (5.118)$$

where the Kalman gain vector is given in (E.16) and the error is defined in (5.10). Using the matrix recursion (E.13), the weight vector update can be expressed as

$$\hat{\mathbf{w}}_M^\ell(n) = \beta \Phi_M^{-1}(n) \Phi_M(n-1) \hat{\mathbf{w}}_M^\ell(n-1) + \Phi_M^{-1}(n) \mathbf{u}_M(n) r^*(n+\ell). \quad (5.119)$$

Convergence in the Mean

Take the expected value on each side of (5.119) and assume, as in [33], that $\Phi_M(n)$ is independent of $\mathbf{u}_M(n)$ and $r^*(n+\ell)$ †. Assume also that, in steady-state, $\Phi_M^{-1}(n) \Phi_M(n-1) \approx \mathbf{I}$. Then

$$E[\hat{\mathbf{w}}_M^\ell(n)] = \beta E[\hat{\mathbf{w}}_M^\ell(n-1)] + E[\Phi_M^{-1}(n)] E[\mathbf{u}_M(n) r^*(n+\ell)]. \quad (5.120)$$

From (5.18), the expected value of the deterministic autocorrelation matrix is

$$\begin{aligned} E[\Phi_M(n)] &= \sum_{i=1}^n \beta^{n-i} E[\mathbf{u}_M(i) \mathbf{u}_M^H(i)] \\ &= \mathbf{R} \sum_{i=1}^n \beta^{n-i} \\ &= \mathbf{R} \frac{\beta^n - 1}{\beta - 1}. \end{aligned} \quad (5.121)$$

† This is an assumption difficult to justify, but its use by Eleftheriou and Falconer leads to useful results [33].

The expected value of the matrix inverse is then [34]

$$E[\Phi_M^{-1}(n)] = \mathbf{R}^{-1} \frac{\beta - 1}{\beta^n - 1}, \quad (5.122)$$

and (5.120) becomes

$$\begin{aligned} E[\hat{\mathbf{w}}_M^\ell(n)] &= \beta E[\hat{\mathbf{w}}_M^\ell(n-1)] + \frac{\beta - 1}{\beta^n - 1} \mathbf{R}^{-1} E[\mathbf{p}_n] \\ &= \beta^n E[\hat{\mathbf{w}}_M^\ell(0)] + \mathbf{R}^{-1} E[\mathbf{p}_n]. \end{aligned} \quad (5.123)$$

Because β is lower or equal to one, the above equation converges to

$$\begin{aligned} \lim_{n \rightarrow \infty} E[\hat{\mathbf{w}}_M^\ell(n)] &= \mathbf{R}^{-1} E[\mathbf{p}_n] \\ &= \mathbf{w}_{\text{opt}} + 1/2 v_{\text{ss}} \mathbf{R}^{-1} \ddot{\mathbf{p}}(D), \end{aligned} \quad (5.124)$$

where equation (4.81) was used and the delay estimator is assumed unbiased.

The weight vector is therefore biased, with a bias vector given by

$$\mathbf{b} = 1/2 v_{\text{ss}} \mathbf{R}^{-1} \ddot{\mathbf{p}}(D), \quad (5.125)$$

as in the joint LMS algorithm.

Convergence in the Mean Square

Rearrange (5.119) as

$$\Phi_M(n) \hat{\mathbf{w}}_M^\ell(n) = \beta \Phi_M(n-1) \hat{\mathbf{w}}_M^\ell(n-1) + \mathbf{u}_M(n) r^*(n+\ell), \quad (5.126)$$

and subtract the vector $\Phi_M(n) \mathbf{w}_{\text{opt}}$ from each side of (5.126), where

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{p}_n |_{\ell=-D}. \quad (5.127)$$

The following update equation for the weight noise vector is then obtained

$$\boldsymbol{\eta}_M(n) = \beta \Phi_M^{-1}(n) \Phi_M(n-1) \boldsymbol{\eta}_M(n-1) + \Phi_M^{-1}(n) \mathbf{u}_M(n) e_o^*(n, \ell), \quad (5.128)$$

where the error is defined as

$$e_o(n, \ell) = r(n+\ell) - \mathbf{w}_{\text{opt}}^H \mathbf{u}_M(n). \quad (5.129)$$

The weight noise correlation matrix is then

$$\begin{aligned} \mathbf{K}_\eta(n) &= E[\boldsymbol{\eta}_M(n) \boldsymbol{\eta}_M^H(n)] \\ &= \beta^2 E[\Phi_M^{-1}(n) \Phi_M(n-1) \boldsymbol{\eta}_M(n-1) \boldsymbol{\eta}_M^H(n-1) \Phi_M(n-1) \Phi_M^{-1}(n)] \\ &\quad + \beta E[\Phi_M^{-1}(n) \Phi_M(n-1) \boldsymbol{\eta}_M(n-1) \mathbf{u}_M^H(n) \Phi_M^{-1}(n) e_o(n, \ell)] \\ &\quad + \beta E[e_o^*(n, \ell) \Phi_M^{-1}(n) \mathbf{u}_M(n) \boldsymbol{\eta}_M^H(n-1) \Phi_M(n-1) \Phi_M^{-1}(n)] \\ &\quad + E[|e_o(n, \ell)|^2 \Phi_M^{-1}(n) \mathbf{u}_M(n) \mathbf{u}_M^H(n) \Phi_M^{-1}(n)]. \end{aligned} \quad (5.130)$$

Using the assumptions leading to (5.120), the second and third terms of (5.130) are approximately zero, because, by orthogonality principles, $E[\mathbf{u}_M(n)e_o^*(n,\ell)] \approx 0$ [33]. The correlation matrix is then of the form

$$\mathbf{K}_\eta(n) \approx \beta^2 \mathbf{K}_\eta(n-1) + E[|e_o(n,\ell)|^2] E[\Phi_M^{-1}(n) \mathbf{R} \Phi_M^{-1}(n)]. \quad (5.131)$$

It is shown in [33] that the last expectation of (5.131) can be written as

$$\begin{aligned} E[\Phi_M^{-1}(n) \mathbf{R} \Phi_M^{-1}(n)] &= E[\Phi_M^{-1}(n) \mathbf{R} \Phi_M^{-1}(n) \mathbf{R}] \mathbf{R}^{-1} \\ &\approx (1-\beta)^2 E[(\mathbf{I} - \mathbf{P}(n))^2] \mathbf{R}^{-1} \\ &\approx (1-\beta)^2 (\mathbf{I} + E[\mathbf{P}^2(n)]) \mathbf{R}^{-1}, \end{aligned} \quad (5.132)$$

where $\mathbf{P}(n)$ is a zero-mean fluctuation matrix that manifests the fluctuations of the product $\Phi_M^{-1}(n) \mathbf{R}$ around the identity matrix \mathbf{I} , and is defined as

$$\mathbf{P}(n) = (1-\beta) \mathbf{R}^{-1} \tilde{\Phi}_M(n), \quad (5.133)$$

where $\tilde{\Phi}_M(n)$ is assumed to be a Hermitian perturbation matrix such that (using equation (5.121))

$$\begin{aligned} \tilde{\Phi}_M(n) &= E[\Phi_M(n)] + \tilde{\Phi}_M(n) \\ &= \mathbf{R} \frac{\beta^n - 1}{\beta - 1} + \tilde{\Phi}_M(n). \end{aligned} \quad (5.134)$$

Note that the entries s_{ij} of the matrix $S = E[\mathbf{P}^2(n)]$ can be computed as [33]

$$\begin{aligned} s_{ij} &\approx \frac{1-\beta}{1+\beta} \left[\sum_{k=1}^M \mathcal{R}_{ij} \text{var}\{u(n-j+1)u^*(n-k+1)\} \mathcal{R}_{kk} \right. \\ &\quad \left. + \sum_{\substack{k=1 \\ k \neq j}}^M \mathcal{R}_{ik} E[(u(n-k+1)u^*(n-j+1) - r_{kj})^2] \mathcal{R}_{jk} \right], \end{aligned} \quad (5.135)$$

where r_{ij} and \mathcal{R}_{ij} represent respectively the entries of \mathbf{R} and \mathbf{R}^{-1} .

The expectation of the error squared in (5.131) is

$$E[|e_o(n,\ell)|^2] = E[\xi_o(\ell)], \quad (5.136)$$

where $\xi_o(\ell)$ is defined in equation (3.39) and the expected value in the right hand side is taken with respect to the delay value. Collecting (5.131), (5.132) and (5.136), the update equation for the correlation matrix is

$$\begin{aligned} \mathbf{K}_\eta(n) &\approx \beta^2 \mathbf{K}_\eta(n-1) + (1-\beta)^2 (\mathbf{I} + E[\mathbf{P}^2(n)]) \mathbf{R}^{-1} E[\xi_o(\ell)] \\ &\approx \beta^{2n} \mathbf{K}_\eta(0) + \frac{1-\beta}{1+\beta} (\mathbf{I} + E[\mathbf{P}^2(n)]) \mathbf{R}^{-1} E[\xi_o(\ell)] (1-\beta^{2n}). \end{aligned} \quad (5.137)$$

Letting n tend to infinity and using equation (3.57), the steady-state weight noise correlation matrix is

$$\mathbf{K}_\eta \approx \frac{1-\beta}{1+\beta} (\mathbf{I} + E[\mathbf{P}^2(n)]) \mathbf{R}^{-1} [\xi_{\min} + 1/2v_{ss} \ddot{\xi}_{\min}]. \quad (5.138)$$

5.5.1.3 Excess Mean-Squared Error and Misadjustment with the Joint LS Algorithm

Proceeding as in Subsection 4.3.1.3, the excess MSE is given by

$$\xi_{ss} = \xi_{\min} + v_{ss}\ddot{\xi}_{\min}/2 + E[\eta_n^T \mathbf{R} \eta_n]. \quad (5.139)$$

The last term of (5.139) is given by

$$E[\eta_n^T \mathbf{R} \eta_n] = \text{tr}[\mathbf{K}_\eta \mathbf{R}], \quad (5.140)$$

which gives, using (5.138),

$$E[\eta_n^T \mathbf{R} \eta_n] = \text{tr}[\mathbf{I} + E[\mathbf{P}^2(n)]] \frac{1 - \beta}{1 + \beta} [\xi_{\min} + 1/2 v_{ss} \ddot{\xi}_{\min}]. \quad (5.141)$$

For Gaussian signals, the trace in (5.141) was computed, in [33], to be

$$\text{tr}[\mathbf{I} + E[\mathbf{P}^2(n)]] \approx M, \quad (5.142)$$

and (5.139) becomes

$$\xi_{ss} = \xi_{\min} + v_{ss}\ddot{\xi}_{\min}/2 + \frac{1 - \beta}{1 + \beta} M [\xi_{\min} + 1/2 v_{ss} \ddot{\xi}_{\min}]. \quad (5.143)$$

Therefore, equation (4.121) applies with $\xi_{\text{ex}}^{\text{d}}$ defined as in (4.122) and

$$\xi_{\text{ex}}^{\text{f}} = \frac{(1 - \beta)M\xi_{\min}}{1 + \beta} \quad (5.144)$$

$$\xi_{\text{ex}}^{\text{df}} = \frac{(1 - \beta)M v_{ss} \ddot{\xi}_{\min}}{2(1 + \beta)}. \quad (5.145)$$

The misadjustment expression is like equation (4.125), i.e.

$$\begin{aligned} \mathcal{M} &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{df}} \\ &= \mathcal{M}^{\text{d}} + \mathcal{M}^{\text{f}} + \mathcal{M}^{\text{d}}\mathcal{M}^{\text{f}}, \end{aligned}$$

where \mathcal{M}^{d} is as in (4.126) and

$$\mathcal{M}^{\text{f}} = \frac{(1 - \beta)M}{1 + \beta} \quad (5.146)$$

and

$$\begin{aligned} \mathcal{M}^{\text{df}} &= \frac{(1 - \beta)M v_{ss} \ddot{\xi}_{\min}}{2(1 + \beta)\xi_{\min}} \\ &= \mathcal{M}^{\text{d}}\mathcal{M}^{\text{f}}. \end{aligned} \quad (5.147)$$

5.5.2 The Joint LS Algorithm in Type I Configuration

The steady-state considerations of Subsection 5.5.1.1 apply in the Type I case and, since the delay is assumed to be transferred to every sample of the adaptive filter line, the above results of the filter analysis are also valid here. Therefore, the excess MSE and misadjustment expressions of Subsection 5.5.1.3 can be used in the study of the joint LS algorithm in Type I configuration.

5.5.3 Discussion

The analyses performed in this section have a goal slightly different from the similar analyses of Chapter 4. In the joint LMS algorithm of the previous chapter, the adaptation factors α and μ influence directly the stability, as well as the steady-state properties of the algorithm (the excess MSE and the misadjustment). The first goal of Section 4.3 is the determination of the ranges of values that both the adaptation factors can take, while producing estimates whose mean and variance are finite in steady-state conditions. The excess MSE and misadjustment expressions are useful in determining the quality of the estimates and follow easily from the stability analysis.

In the present section, there is no such stability ranges, since the LS algorithm is inherently stable, when infinite precision arithmetic is used. The weighting factor β influences the convergence speed and the precision of the estimation, and its range of value is usually between 0.9 and 1.0. The goal of this section was therefore to determine the quality of the joint estimation, by deriving excess MSE and misadjustment expressions. This is why the discussion about the delay estimate mean and variance, performed in Subsection 5.5.1.1, is only qualitative. The analysis of the adaptive filter given in Subsection 5.5.1.2 is mainly useful in the computation of the excess MSE. Note however that the expressions obtained for the mean and correlation matrix of the weight vector are similar to those obtained in Chapter 4. In particular, the weight vector is biased by the same vector in both cases and both the correlation matrices are functions of the expression $[\xi_{\min} + 1/2v_{ss}\ddot{\xi}_{\min}]$ (compare equations (4.110) and (5.138)). Note also that $\xi_{\text{ex}}^{\text{f}}$ has again a form identical to the form for a filter operating alone [33].

As for the expressions (5.143) and (5.147), they show again that the misadjustment is a function of three terms, one more specific to the adaptive delay, one related to the adaptive filter and finally one equal to the product of the first two terms.

5.6 Summary

Joint time delay estimation and adaptive RLS filtering, using a fast transversal filter implementation, has been considered in this chapter. The philosophy adopted here was fairly different than the orientation of the previous chapters, since the most part of the sections was devoted to the derivation and description of a new form of LS algorithm. This joint delay estimation and LS adaptive filtering algorithm allows the efficient computations of the current optimum weight vector, and of the optimum integer delay (lag).

A set of lag-recursive relations was derived geometrically, for the computation of both the LS weight vector solution and the minimum sum of squared errors. These relations are functions of the same internal variables used in the fast transversal adaptive filter, and the lag-recursive relations are appended to a form of FTF algorithm, to produce the joint LS algorithm. The order of the predictors used in the FTF algorithm must be $M - 1$, if the adaptive filter order is M . The lag-recursive relations were also used to derive a lag-update algorithm, which was used to adapt the integer delay estimator.

The delay estimate behaviour was considered qualitatively and the steady-state weight error correlation matrix was derived. Finally, the excess MSE and misadjustment were found to be functions of the term $[\xi_{\min} + 1/2v_{\text{ss}}\ddot{\xi}_{\min}]$, as in the joint LMS algorithm.

The material presented in this chapter is mainly theoretical, although the final joint LS algorithm of Section 5.4.1 can be implemented as such. More practical considerations are given in the next chapter where numerous simulation results are given.

Chapter 6

Experimental Results: The Joint LMS Algorithm and the Joint RLS Algorithm

6.1 Introduction

So far, the work presented in this thesis has been analytical. Chapter 3 served the purpose of investigating the theoretical behaviour of the joint steepest-descent algorithm. In particular, the possibility of convergence to a multitude of stationary points has been demonstrated. The role of the second derivative of the MSE function, in the stability of the delay tracking portion of the joint algorithm, was derived. Some bounds, useful in the practical application of the joint SD algorithm, were derived and discussed. In the present chapter, the properties of the SD algorithm are illustrated with practical examples and the stability bounds are computed.

The joint LMS algorithm was presented in Chapter 4 as a stochastic implementation of the joint SD algorithm. Its analysis was performed for joint convergence in the mean and in the mean square. Some theoretical bounds on the two gain factors involved in the algorithm were derived and the expressions for the excess MSE and the misadjustment of the joint algorithm were obtained. A design procedure, for the determination of the two gain factors, was presented. The bounds and the excess MSE are computed in the following sections, and the critical parameters used in the design procedure are illustrated.

In Chapter 5, the focus was given to the derivation of some lag-recursive relations and to the definition of a new form of RLS algorithm. The joint algorithm is fairly complicated and no theoretical study was performed about its behaviour. The expressions for the excess MSE

and the misadjustment were obtained. The joint RLS algorithm is implemented integrally as derived and its practical behaviour is studied in the actual chapter.

This chapter is therefore structured as follows. In Section 6.2, an experimental set-up is defined for the simulations of the joint algorithms. In particular, the reference filter that is used in most of the simulations is described, and the implementation of the algorithms is discussed. Then the results of Chapters 3 and 4 are investigated in Section 6.3, and the joint RLS algorithm is simulated in Section 6.4. A hybrid joint algorithm is briefly discussed in Section 6.5.2. This algorithm is made of an LMS adaptive delay algorithm and an RLS adaptive filter algorithm.

6.2 Experimental Set-Up

All the simulations were implemented in a system identification (cancellation) configuration (see Figs. 2.6, 2.8 and 3.2). Unless it is otherwise specifically noted, the noiseless input signal $s(n)$ is a zero mean and white Gaussian process, as are the two noise sources. All the signals and systems are real.

Unless otherwise noted, the reference filter is a 21-tap lowpass transversal filter, with a 3dB bandwidth approximately equal to 0.7π . Its impulse response and its transfer function are illustrated in Figs. 6.1 and 6.2. This choice is somewhat arbitrary and is dictated by the ease the filter can be implemented in the actual simulations. Some results with a more realistic filter are presented in Sections 6.5.1 and 6.5.2.

The reference filter can be made time-varying by changing its amplitude and/or phase response with time. A very specific reference filter nonstationarity is simulated. The variations of the filter amplitude and phase responses are constant over the whole filter frequency range. This implies that no frequency selective nonstationarity is applied and that the reference transfer function is of the form

$$R(e^{j\omega}) = H(e^{j\omega})A(n)e^{j\theta(n)},$$

where $H(e^{j\omega})$ is the *stationary* reference filter transfer function and $A(n)e^{j\theta(n)}$ is a frequency independent time-varying gain.

The cases simulated are for linearly and sinusoidally varying amplitude and phases of the form

$$\begin{cases} A(n) = 1 + f(n) \\ \theta(n) = f(n)\pi/2, \end{cases}$$

with

$$f(n) = S \cdot n \quad S = \text{slope},$$

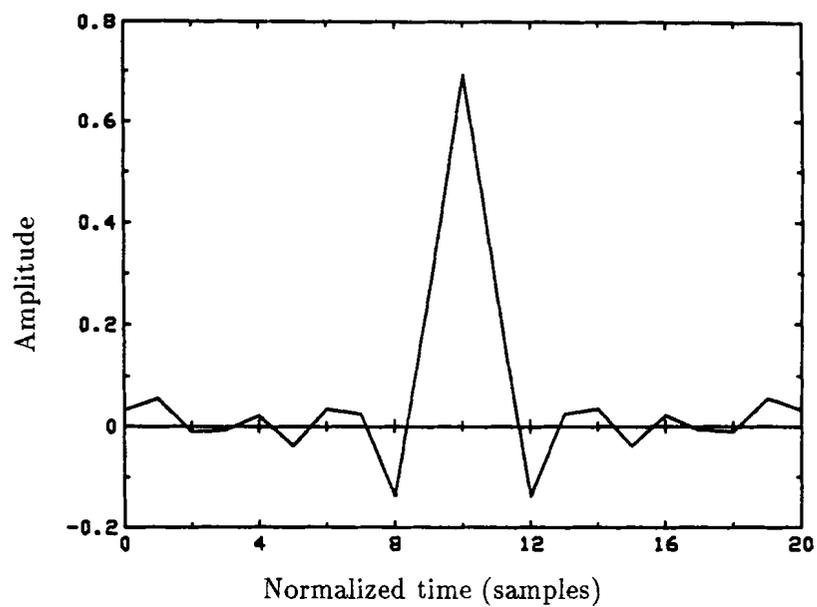


Fig. 6.1 Reference filter impulse response $h(n)$

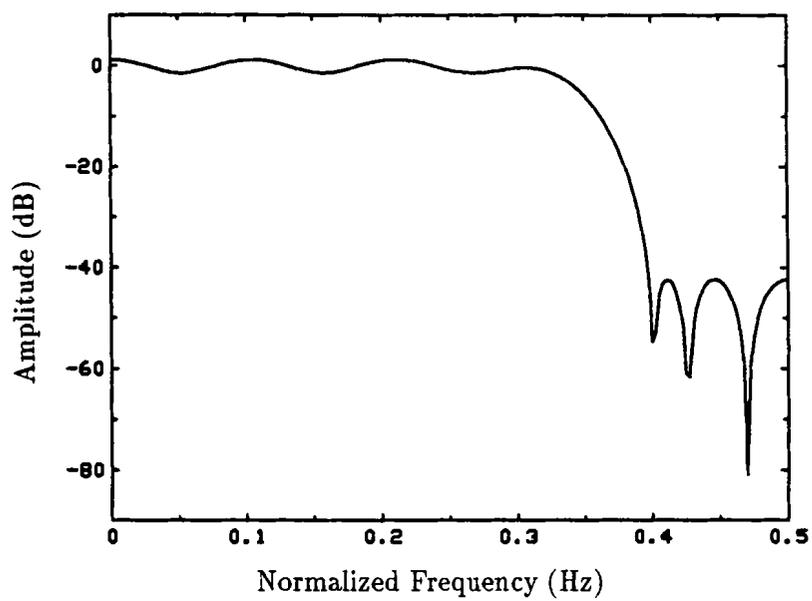


Fig. 6.2 Reference filter transfer function $H(e^{j\omega})$

or

$$f(n) = \sin(2\pi n/P) \quad P = \text{period.}$$

Note that when both the amplitude and the phase are time-varying, they experience the same kind of nonstationarity (linear or sinusoidal).

The delays are implemented as follows. Consider a sequence $s(n)$ and its delayed version $s(nT - D)$, where D is a constant. It is desired to obtain $s(nT - D)$ by passing $s(n)$ through a time-invariant filter whose impulse response is [67]

$$\tilde{g}_d(n) = \frac{\sin \pi(n - D/T)}{\pi(n - D/T)}. \quad (6.1)$$

This impulse response is infinite in time and must be truncated and delayed if it is to be implemented as a causal transversal digital filter. Since the function $\tilde{g}_d(n)$ approaches zero as n increases, the truncation can take place with minimal effects [67], [26]. It is also shown experimentally in [67] that the modelling error is largest at $D/T = 0.25$ and that it is lower than 1 percent for an impulse response in excess of 60 weights.

Therefore, the *fractional* part of both the adaptive delay d_n and the reference delay D_n are implemented using a delayed 75-tap version of (6.1), i.e.

$$g_d(n) = \frac{\sin \pi(n - 37 - D/T)}{\pi(n - 37 - D/T)} \quad 0 \leq n \leq 74. \quad (6.2)$$

In order to allow for integer delays, the shift register on which $g_d(n)$ is applied has a length N larger than 75. By sliding the 75-tap impulse response along the shift register, an overall delay of $A + D/T$ samples can be obtained, where A is an integer number comprised between zero and $N - 75$, and D/T is a rational number lower than one. The delay of 37 samples introduced by $g_d(n)$ is fixed and is taken into consideration in the simulations.

The adaptive negative delay $-d_n$, present in the reference branch of the Type II-DRB cancellation configuration (Fig. 3.2), is implemented by applying a fixed delay D_f on the adaptive filter input signal $u(n)$ and by redefining the adaptive delay as $D_f - d_n$.

6.3 Results with The Joint LMS Algorithm

The first part of this section is devoted to a discussion about the simulation implementation. Then the general results obtained in Chapter 3 for the joint SD algorithm, and their application in the joint LMS algorithm are considered. The specific results of Chapter 4 are investigated in Subsection 6.3.7.

6.3.1 Simulation of the LMS Algorithm

The joint LMS algorithm in Type I configuration, given in equations (4.32) and (4.33), is simulated according to the blockdiagram of Fig. 6.3. The derivative of the adaptive filter output, with respect to the delay d_n , is given by

$$\dot{y}(n, d_n) = \mathbf{w}_n^T \dot{\mathbf{u}}(nT - d_n). \quad (6.3)$$

It is implemented by passing the delayed input signal derivative through a replica of the adaptive filter. This derivative can be obtained from $u(n)$ with a filtering operation. The following development, analog to the one performed in [13], leads to the derivative filter impulse response.

The *continuous* signal $u(t)$ can be obtained from the sequence $u(n)$ by the interpolation operation [20]

$$u(t) = \sum_n u(n) \frac{\sin \pi(t - nT)/T}{\pi(t - nT)/T}. \quad (6.4)$$

The derivative of $u(t)$ with respect to t is then

$$\frac{\partial u(t)}{\partial t} = \sum_n u(n) \left[\frac{\cos \pi(t - nT)/T}{t - nT} - \frac{\sin \pi(t - nT)/T}{\pi(t - nT)^2/T} \right], \quad (6.5)$$

and the derivative of $u(t)$ with respect to d_n is

$$\frac{\partial u(t)}{\partial d_n} = \frac{\partial u(t)}{\partial t} \frac{\partial t}{\partial d_n}. \quad (6.6)$$

Therefore, using (6.5) and (6.6),

$$\begin{aligned} \frac{\partial u(nT - iT - d_n)}{\partial d_n} = \\ - \sum_j u(j) \left[\frac{\cos \pi(nT - iT - jT - d_n)/T}{nT - iT - jT - d_n} - \frac{\sin \pi(nT - iT - jT - d_n)/T}{\pi(nT - iT - jT - d_n)^2/T} \right]. \end{aligned} \quad (6.7)$$

Equation (4.33) can then be implemented as

$$d_{n+1} = d_n - 2\alpha\epsilon(n) \sum_{i=0}^L w_{ni} q(n-i), \quad (6.8)$$

where $q(n)$ is the output of the derivative filter with impulse response

$$\bar{b}_d(n) = \frac{\cos \pi(nT - d_n)/T}{nT - d_n} - \frac{\sin \pi(nT - d_n)/T}{\pi(nT - d_n)^2/T}. \quad (6.9)$$

As with the delay elements simulation, this impulse response has to be truncated and delayed in order to obtain a causal filter response. The truncation window is again of length 75 and the derivative filter is implemented with weights

$$b_d(n) = \frac{\cos \pi(n - 37 - d_n/T)}{T(n - 37 - d_n/T)} - \frac{\sin \pi(n - 37 - d_n/T)}{T\pi(n - 37 - d_n/T)^2} \quad 0 \leq n \leq 74. \quad (6.10)$$

By assuming that the sampling period is $T = 1$, both the impulse responses $g_d(n)$ and $b_d(n)$ can be easily adapted to the variations of d_n .

The Type II-DAB and Type II-DRB configurations can be implemented in a similar way by applying the derivative filter directly on the adaptive filter output or on the reference signal. Note the difference between the Type I and Type II implementations. In the former, the derivative filter being located before the adaptive filter replica, the derivative applies only to one sample in the filter delay line, as does the adaptive delay in the adaptive branch. In the latter, the derivative being taken on the adaptive filter output, all the samples of the delay line are implicitly derived.

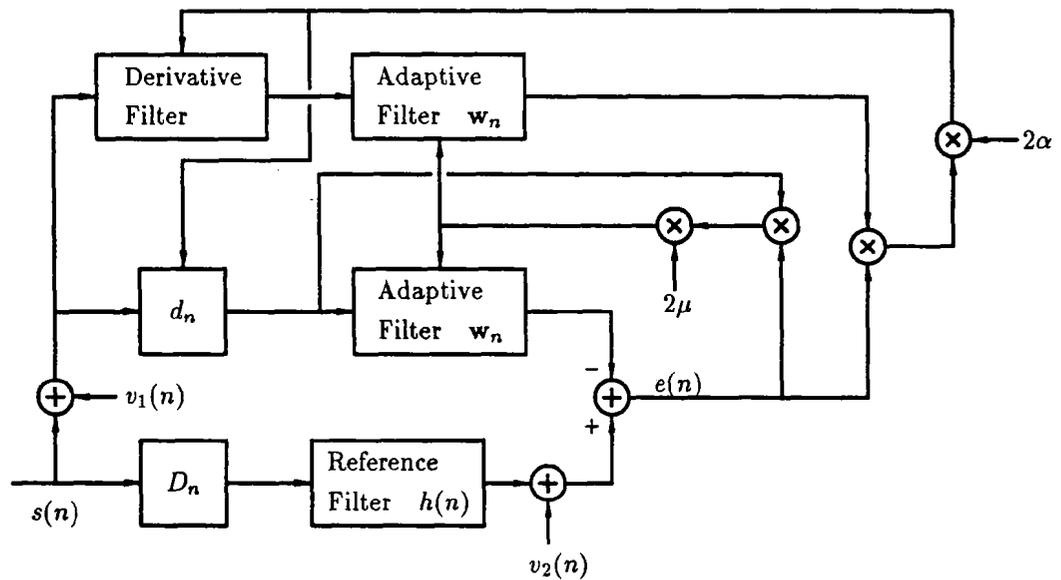


Fig. 6.3 Blockdiagram of the simulation of a Type I configuration

The systems parameters needed to apply the analytical results of the previous chapters are obtained as follows. The deterministic autocorrelation corresponding to the reference filter of Fig. 6.1 is shown in Fig. 6.4. The value $\rho_h(0)$ corresponds to the maximum in this figure. From this function, the second and fourth derivatives $\rho_h''(0)$ and $\rho_h^{(4)}(0)$ can be found. These values are

$$\begin{cases} \rho_h(0) = 0.6661 \\ \rho_h''(0) = -0.9753 \\ \rho_h^{(4)}(0) = 2.6508. \end{cases} \quad (6.11)$$

The minimum MSE ξ_{\min} and its second and fourth derivative are also necessary in the application of the results of Chapters 3 and 4. The MMSE is given by equation (4.194) and

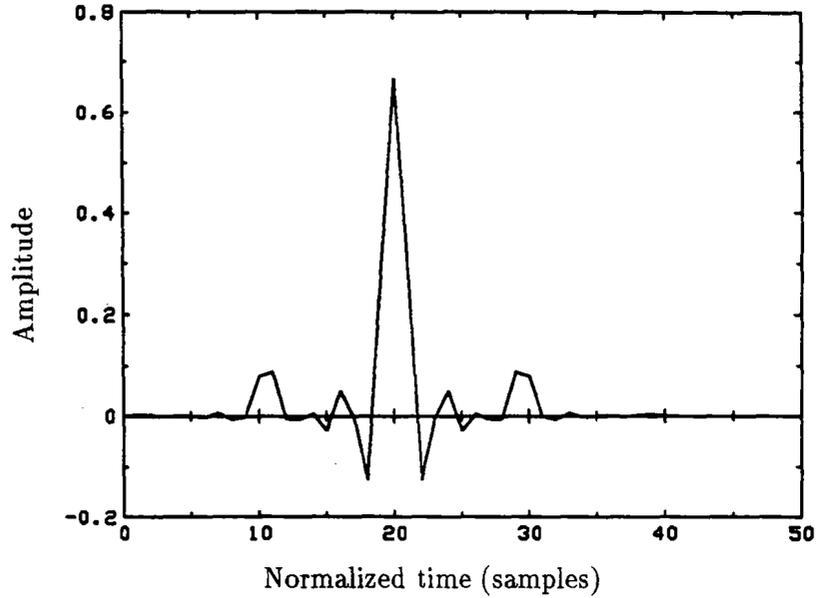


Fig. 6.4 Reference filter deterministic autocorrelation function $\rho_h(n)$

for white signals and equal noise variances this equation can be expressed as

$$\xi_{\min} = \sigma_{v_1}^2 [1 - \rho_{w_{\text{opt}}}(0)] + \Phi_{ss} [\rho_h(0) - \rho_{w_{\text{opt}}}(0)], \quad (6.12)$$

where $\rho_{w_{\text{opt}}}(k)$ is the deterministic autocorrelation of the optimum filter, for a given signal-to-noise ratio. This optimum filter is given in equation (3.84) for the cancellation scenario considered in this chapter. Combining (3.84) and (6.12), the following expression for the MMSE is obtained

$$\xi_{\min} = \sigma_{v_1}^2 \left[1 - \frac{\rho_h(0)}{(1 + 1/\text{SNR}_1)^2} \right] + \Phi_{ss} \rho_h(0) \left[1 - \frac{1}{(1 + 1/\text{SNR}_1)^2} \right], \quad (6.13)$$

where SNR_1 is defined as

$$\text{SNR}_1 = \frac{\Phi_{ss}}{\sigma_{v_1}^2}. \quad (6.14)$$

The second derivative is given in equation (3.115) as

$$\ddot{\xi}_{\min} = \frac{-2\Phi_{ss}^2 \rho_h''(0)}{\Phi_{ss} + \sigma_{v_1}^2}, \quad (6.15)$$

and the fourth derivative can be derived in the same way as

$$\xi_{\min}^{[4]} = \frac{-2\Phi_{ss}^2 \rho_h^{(4)}(0)}{\Phi_{ss} + \sigma_{v_1}^2}. \quad (6.16)$$

The derivative of the input signal is also necessary for the application of the results of Chapter 4. For white signal processes, it can be derived to be

$$\phi_{uu}^{(i)}(0) = \left(\frac{j}{T}\right)^i \frac{(\Phi_{ss} + \sigma_{v_1}^2)\pi^i}{i+1} \quad \text{for } i \text{ even,} \quad (6.17)$$

where $j = \sqrt{-1}$ and the sampling period is taken to be one. Finally, unless otherwise noted, the input signal power spectral density is

$$\Phi_{ss} = \frac{1}{12}, \quad (6.18)$$

which implies that the maximum eigenvalue of the input signal autocorrelation matrix is

$$\begin{aligned} \lambda_{\max} &= \Phi_{ss} + \sigma_{v_1}^2 \\ &= 1/12 + \sigma_{v_1}^2. \end{aligned} \quad (6.19)$$

6.3.2 Multiple Convergence Points and Excess MSE

The presence of multiple convergence points is first illustrated. The reference delay is fixed at a certain value and the adaptive filter is allowed to adapt to this condition, while the adaptive delay is frozen ($\alpha = 0.0$). The optimum weight vector is then obtained for the reference delay fixed at 0, 0.5, 1.0 and 1.5 samples and the MSE function ξ_n is measured, as a function of the relative delay $D_n - d_n$, using these different weight vectors. The results are given in Figs. 6.5 and 6.6. It is first noted that the MSE function exhibits a well defined minimum at $d_n = 0$, for each case. This shows that the condition $\nabla_{\mathbf{w}} \xi_n = 0$ implies $\partial \xi_n / \partial d_n = 0$, as pointed out in Subsection 3.3.1. Furthermore, each of these minimum corresponds to the function $\xi_o(d_n)$, with $d_n = 0$, defined in equation (3.39) as

$$\xi_o(d_n) = \xi_n |_{\mathbf{w}_n = \mathbf{R}^{-1} \mathbf{p}_n}. \quad (6.20)$$

The value of the MSE function at each of these minimums corresponds to the excess MSE defined in equation (3.58). Note that, in none of these cases can the excess MSE be approximated by equation (3.61), because the relative delay is too large.

6.3.3 Delay Tracking Bounds

As derived in Chapter 3, the stability bounds involved in the joint SD algorithm are functions of the quantity $\ddot{\xi}_{\min}$. For the white signals case, $\ddot{\xi}_{\min}$ is given in (6.15). Using (6.11) and (6.18), Table 6.1 can be computed, where α_{\max} is defined as

$$\alpha_{\max} = \frac{2}{\ddot{\xi}_{\min}}. \quad (6.21)$$

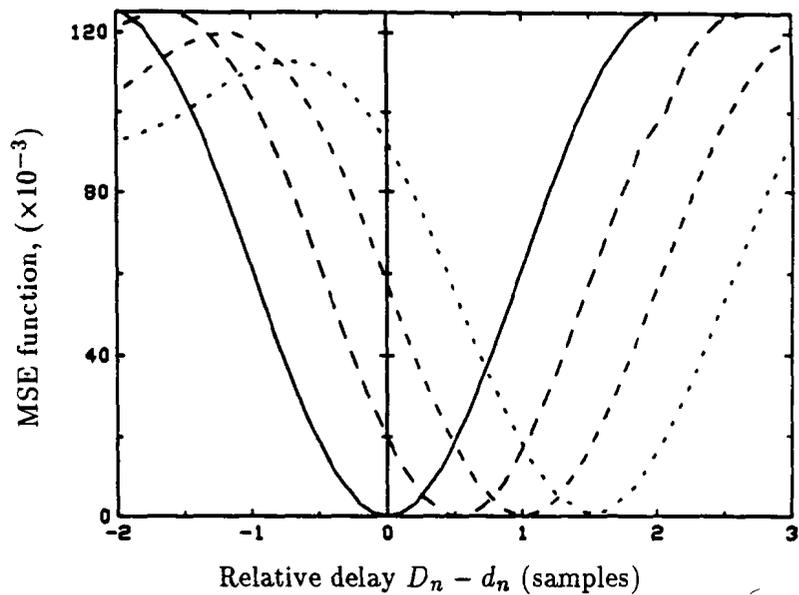


Fig. 6.5 The MSE function for different fixed reference delays D_n ; continuous curve: $D_n = 0.0$, large dashes curve: $D_n = 0.5$, medium dashes curve: $D_n = 1.0$, small dashes curve: $D_n = 1.5$

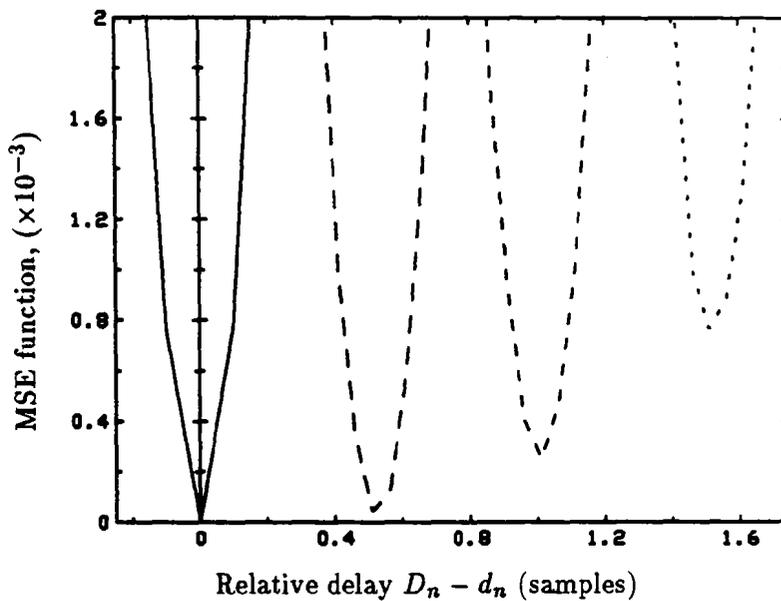


Fig. 6.6 Expanded view of Fig. 6.5

SNR ₁ (dB)	$\ddot{\xi}_{\min}$	α_{\max}
0	0.0813	24.6
10	0.1478	13.53
20	0.1609	12.43
30	0.1624	12.32
∞	0.1626	12.3

Table 6.1 Values of $\ddot{\xi}_{\min}$ and $\alpha_{\max} = 2/\ddot{\xi}_{\min}$ for different signal-to-noise ratios

Note that, because white signals are used, the bound (3.82) of Proposition 3.2 is equal to α_{\max} for infinite SNR. Note also that this value of α corresponds to a safe upper bound, since all other values are superior to it for finite SNR's. This value is also used in Proposition 3.3, in order to define a range of values for alpha such that the adaptive delay is five times faster than the adaptive filter. The range of values, determined with equations (3.92) and (6.19), is illustrated in Fig. 6.7 as a function of μ . The computations were performed for a SNR of 0 dB and for an infinite SNR. The allowable range for α is to the left of the dashed curves and below the continuous curve. Note that for high SNR's, Proposition 3.3 states that α should be larger than 1.0, when $\mu = 0.1$ and that a value $\alpha = 0.1$ is sufficient when $\mu = 0.01$.

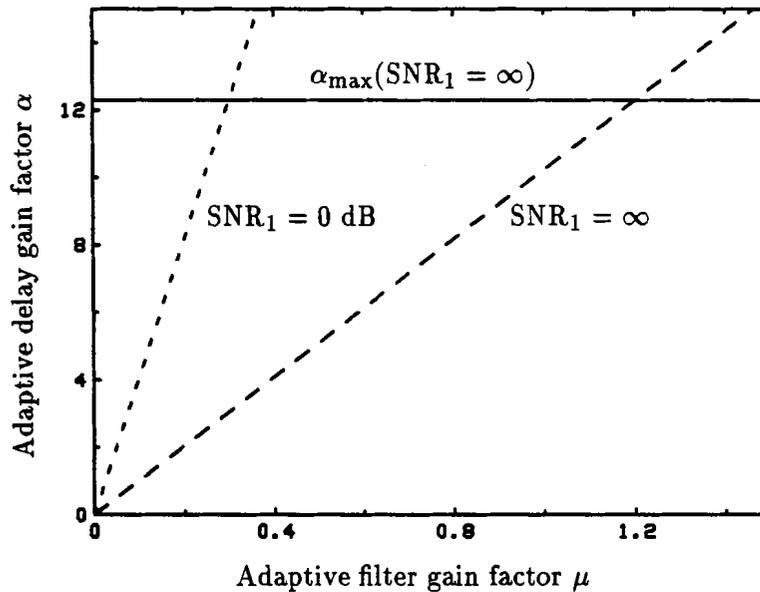


Fig. 6.7 Range of α satisfying Proposition 3.3

6.3.4 Delay Tracking Simulations in Type I

For the joint LMS algorithm in Type I configuration, Proposition 4.5 (equation (4.108)) states that a condition on μ , for convergence in the mean square in noiseless conditions, is (all the eigenvalues are equal to λ_{\max})

$$0 < \mu < \frac{1}{\sum_{i=1}^{21} 1/12} = 0.57. \quad (6.22)$$

It is found experimentally that μ should be below 0.4 for convergence of the adaptive filter in noiseless conditions. This is well below the bound for convergence in the mean established in Proposition 4.4 (equation (4.84)), which indicates that μ should be lower than $1/\lambda_{\max} = 12$.

Similarly, the theoretical bound for convergence in the mean of the adaptive delay, in noiseless conditions, is found to be much larger than the bound found in practice. The theoretical α_{\max} given in Table 6.1 is 12.3, while it is found experimentally that an α superior to 0.9 makes the algorithm unstable in noiseless conditions. These experiments indicate that for μ 's larger than 0.1, it is not possible for α to meet the lower bound established in Fig. 6.7 and still produce a stable algorithm.

6.3.4.1 Adaptive Delay Response to a Reference Delay Step

Based on these results, four combinations of α and μ are first simulated, when a unit delay step is applied in the reference branch. Note that white signals and noiseless conditions are assumed. The results are given in Figs. 6.8 to 6.11. Figs. 6.8 and 6.9 illustrate cases where the lower bound of Proposition 3.3 is not respected. In both cases, the adaptive delay element has a time constant too large to allow close tracking of the reference delay variations. For a fairly large adaptive delay gain factor, Fig. 6.8 shows that the behaviour of the delay adaptation algorithm is that of a higher order system. This implies that the first order approximation made in equation (3.24), based on the truncation of the Taylor expansion of equation (3.21), is not totally right in this case. When α is well within the bound of Proposition 3.3, as in Fig. 6.10, the adaptive delay element follows closely the reference delay. Note the higher variance in the delay value when α is larger. Finally, Fig. 6.11 illustrates a smooth delay adaptation case.

It was established in Proposition 3.4 that a reference delay step of one sample constitutes a safe upper bound for adequate delay tracking of such variation. This bound was determined from the width of the MSE function around its minimum. On Fig. 6.5, it is seen that the main lobe width is on the order of 4 samples, i.e. twice as wide as the width used in Proposition 3.4. It is therefore expected that the adaptive delay can cope, in the actual

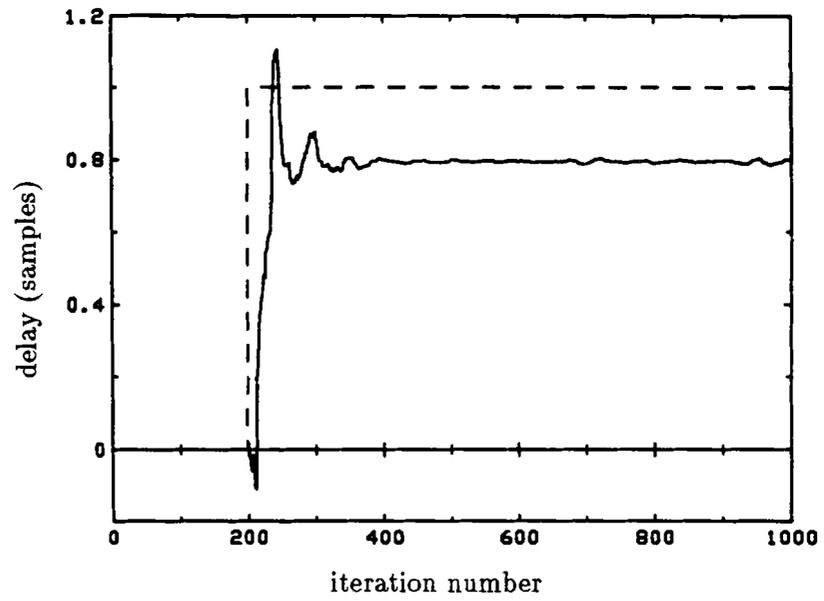


Fig. 6.8 LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.1$, $\alpha = 0.5$

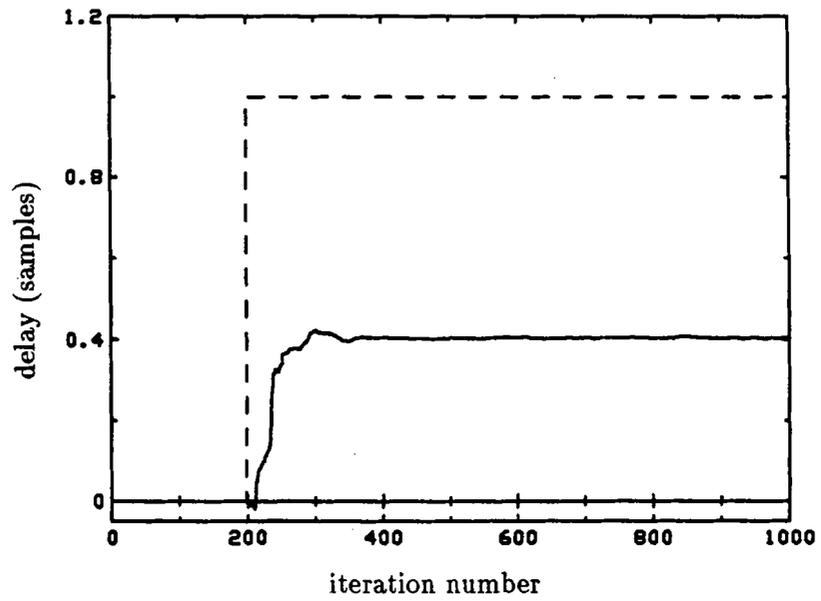


Fig. 6.9 LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.1$, $\alpha = 0.1$

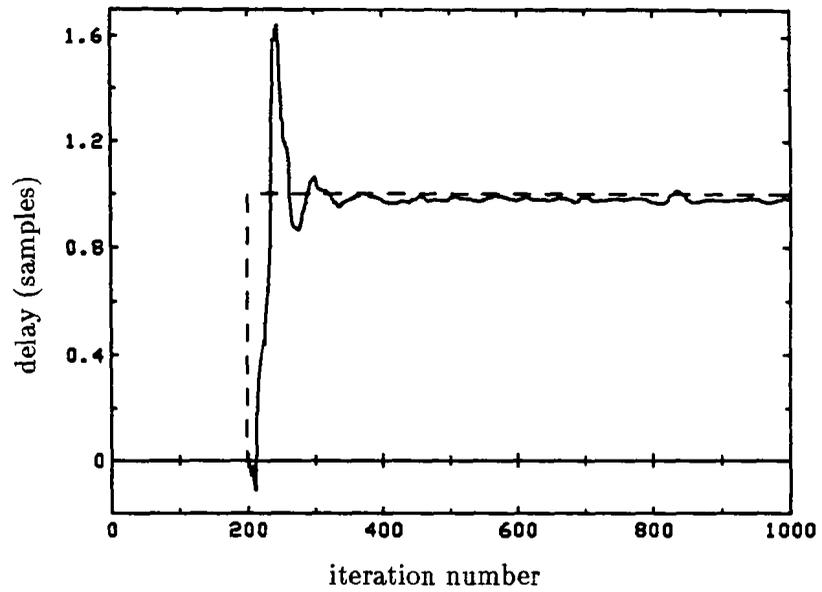


Fig. 6.10 LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

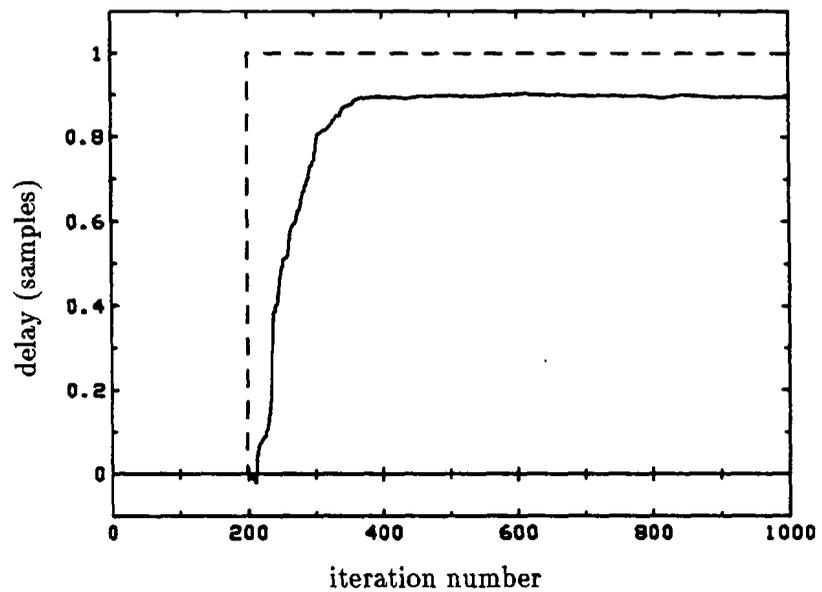


Fig. 6.11 LMS Adaptive delay response to a reference delay unit step; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.1$

simulation, with a reference delay step of 2 samples. The response of the delay estimator, for five different reference delay steps, is shown in Fig. 6.12. As long as the reference delay is within 2 samples, the delay tracking is indeed adequate. But for a step of 2.2 samples, the tracking is less accurate and the time constant is significantly larger. This last behaviour is due to the decrease in the MSE second derivative, as the operating point of the algorithm gets further away from the global minimum.

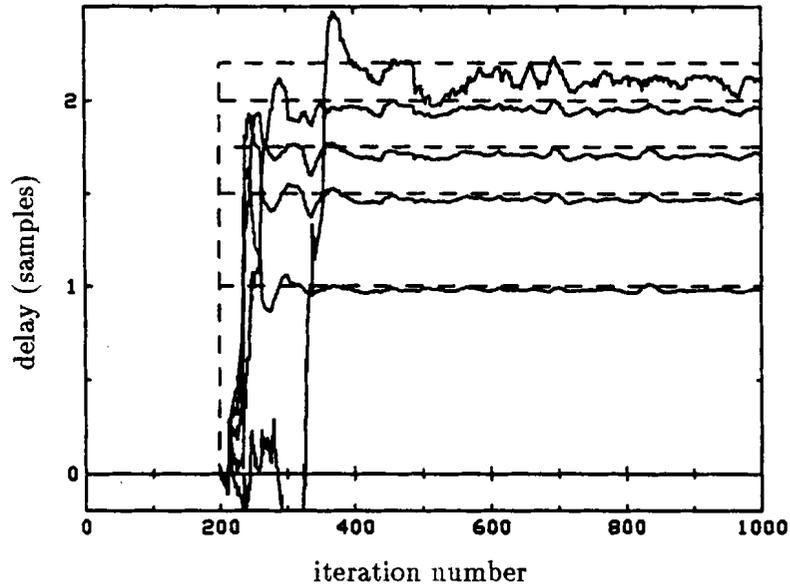


Fig. 6.12 LMS Adaptive delay response to different reference delay step; dashed curves: reference delays; $\mu = 0.01$, $\alpha = 0.5$

From equation (3.76), it is seen that the time constant of delay adaptation is given by

$$\tau_{\text{del}} \approx \frac{1}{\alpha \ddot{\xi}_{\min}}. \quad (6.23)$$

Using the value of $\ddot{\xi}_{\min}$ for infinite SNR, the time constant is on the order of 12 samples for $\alpha = 0.5$ and around 60 samples for $\alpha = 0.1$. These figures are largely confirmed by Figs. 6.10 and 6.11. The learning curves, corresponding to these two figures, are shown in Figs. 6.13 and 6.14. These curves were obtained by averaging 10 different error curves.

Since Proposition 3.3 is true in these cases, the error curve is mainly influenced by the delay adaptation. The time constants of the learning curves is therefore approximately equal to the delay time constant. Fig. 6.13 shows a time constant approximately equal to 15 samples, while the time constant in Fig. 6.14 is on the order of 60 samples. These results confirm the figures computed above with the help of (6.23).

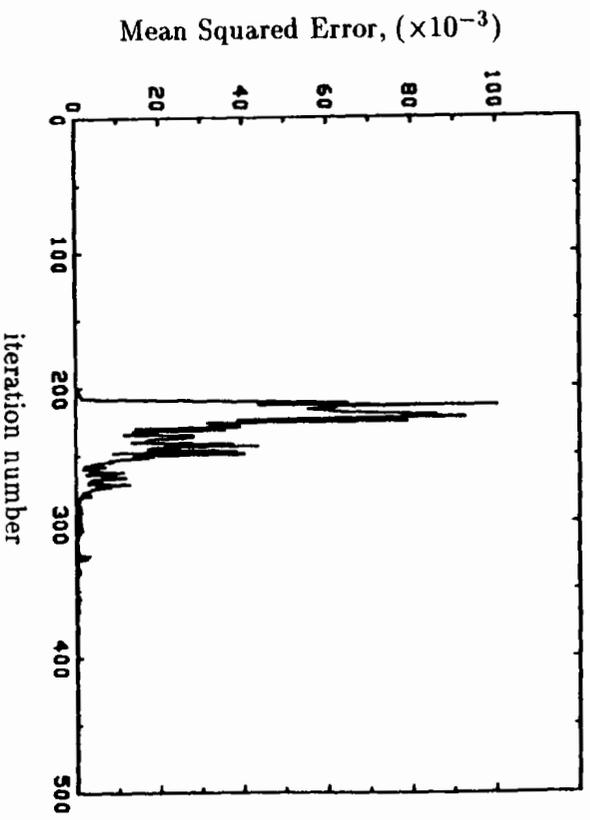


Fig. 6.13 Learning curve for a reference delay unit step; $\mu = 0.01$ and $\alpha = 0.5$

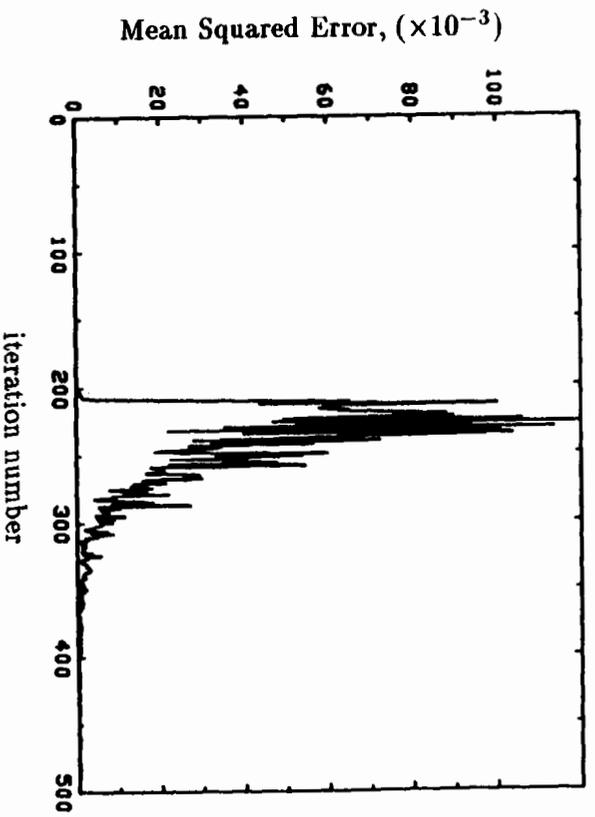


Fig. 6.14 Learning curve for a reference delay unit step; $\mu = 0.01$ and $\alpha = 0.1$

6.3.4.2 Adaptive Delay Response to a Reference Delay Ramp

In processing an audio surveillance tape, it was found in [12] that an adaptive noise canceller can face both linearly and sinusoidally changing reference delays. These variations are essentially caused by the differences in the rotating speed of the recording devices used in the surveillance and in the processing.

The adaptive delay responses to a linearly changing reference delay are presented in Figs. 6.15 and 6.16. The reference slope is 0.01 sample/sample, exceeding the linear variations measured in [12]. This slope is also well below the upper bound on the maximum allowable value computed using Proposition 3.5. Fig. 6.15 illustrates the case where the adaptation speed constraint of Proposition 3.3 is satisfied. The delay element is seen to track very well the delay reference variations. When the constraint is not satisfied, a fraction of the delay variations is compensated for by the adaptive filter, which causes an increasing error between the adaptive and the reference delays, as shown in Fig. 6.16. Note also that in this particular case, the adaptive filter cannot track properly such a rapid reference delay variation and the joint algorithm does not perform satisfactorily after 2000 iterations. The corresponding learning curve is shown in Fig. 6.17.

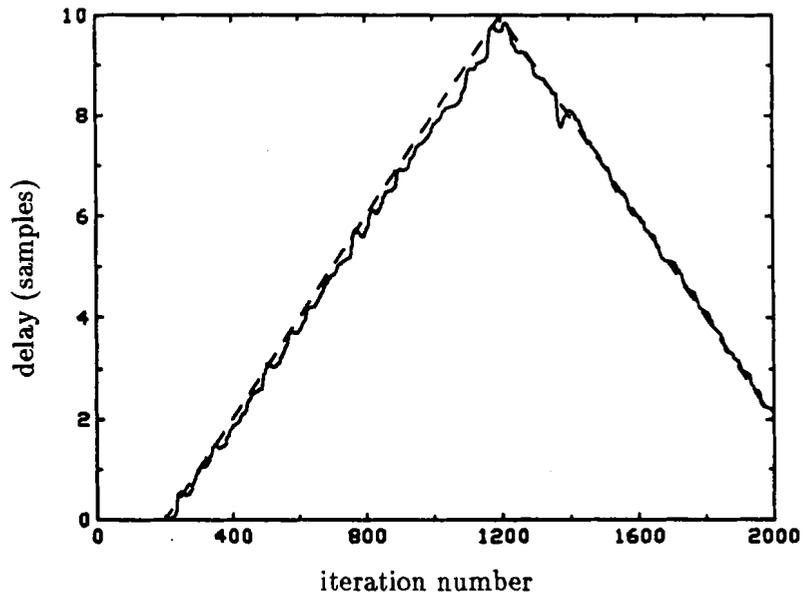


Fig. 6.15 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

6.3.4.3 Adaptive Delay Response to Sinusoidal Reference Delay Variations

The maximum amplitude and period of the sinusoidal variations that can be tracked

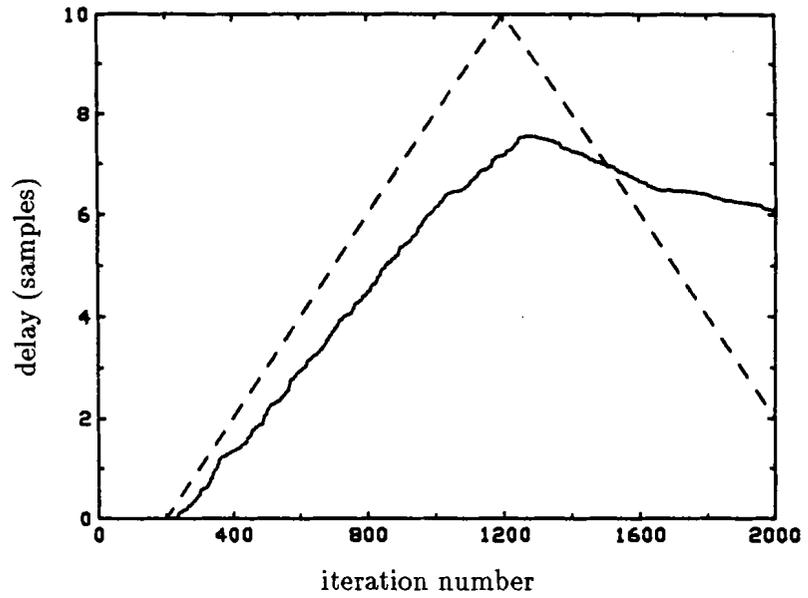


Fig. 6.16 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.1$

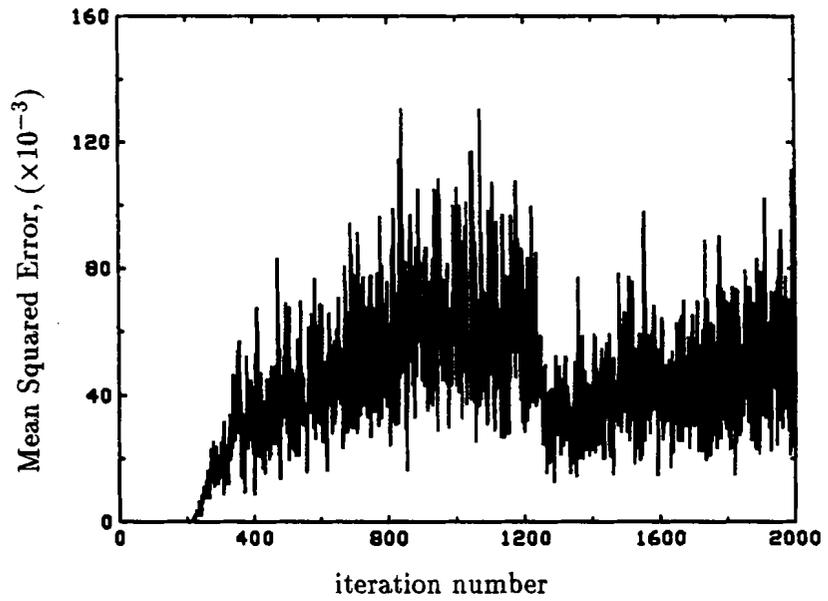


Fig. 6.17 Learning curve for a reference delay ramp of 0.01 sample/sample; $\mu = 0.01$ and $\alpha = 0.1$

are functions of the time constant of adaptation (see Subsection 3.3.2.2). Furthermore, it is argued in the same subsection that, as long as the adaptive delay has a much smaller time constant than the adaptive filter, the former tracks closely the sinusoidal variations if

$$|e^{j2\pi/P} + \alpha\ddot{\xi}_{\min} - 1| \approx \alpha\ddot{\xi}_{\min}.$$

For $\alpha = 0.5$ and $\ddot{\xi}_{\min} = 0.1626$ ($\text{SNR}_1 = \infty$ in Table 6.1), the above approximation is precise to 1% if the period P is about 500 samples, and to 0.27% if the period is 1000 samples. Figs. 6.18 and 6.19 illustrate the delay tracking for these two cases. Note that the tracking is slightly better for the 1000 period case, because the maximum rate of reference delay variations is smaller. Fig. 6.20 illustrates the case where some of the reference delay variations are compensated by the adaptive filter. The resulting adaptive delay response shows a reduced amplitude and a phase lag with respect to the reference.

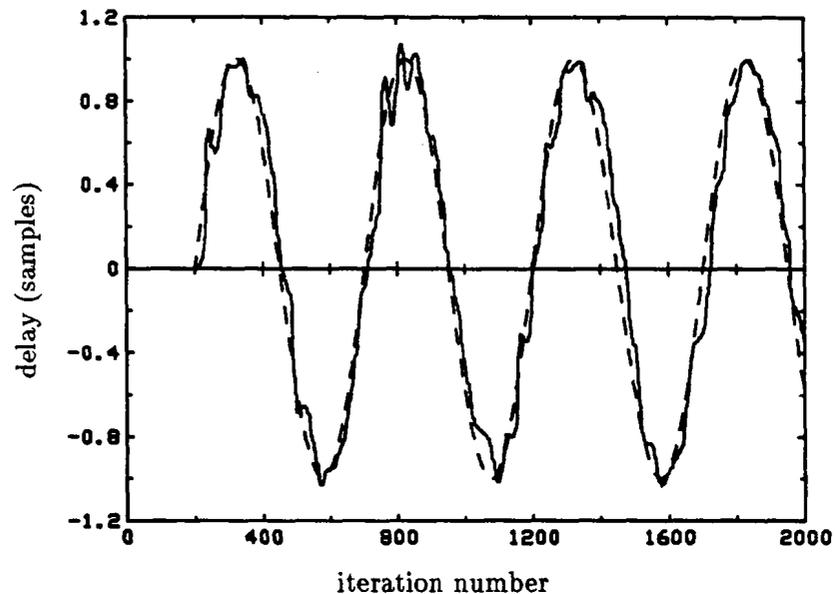


Fig. 6.18 LMS Adaptive delay response to a sinusoidal reference delay variation, period = 500 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

6.3.4.4 Adaptive Delay Response in Noisy Conditions

The above simulation results were obtained in noiseless conditions and show the delay tracking ability of the joint algorithm. When noise is present, the delay estimation is less accurate and the variance of the estimator is increased. This is illustrated in Figs. 6.21 to 6.23, for the three types of reference delay variations considered above. The signal-to-noise

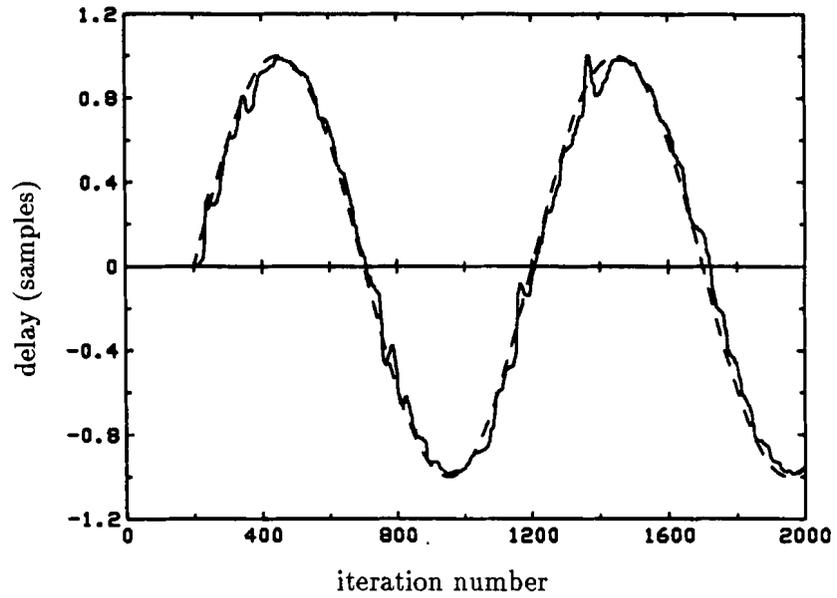


Fig. 6.19 LMS Adaptive delay response to a sinusoidal reference delay variation, period = 1000 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

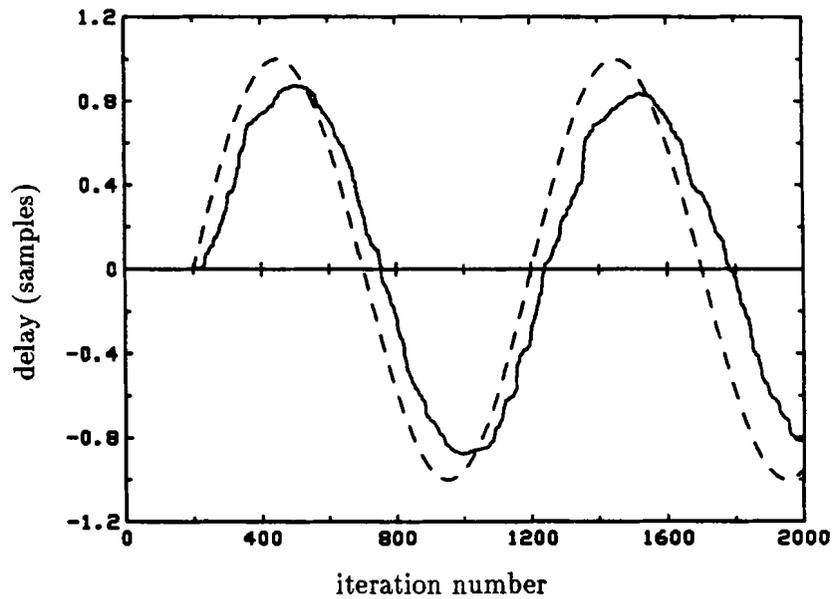


Fig. 6.20 LMS Adaptive delay response to a sinusoidal reference delay variation, period = 1000 samples, amplitude = 1 sample; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.1$

ratio was 10 dB in each of the two noise sources present in the system. The delay tracking is seen to be satisfying, even for this fairly low SNR. The degradation for lower SNR's is gentle, and the delay tracking still takes place at 0 dB.

6.3.4.5 Adaptive Delay Response with a Nonstationary Reference Filter

The purpose of the adaptive filter is to track the variations in the reference filter. In audio surveillance tape analysis, it is likely that these variations are slow, as noticed in [12]. Therefore, a gain factor μ on the order of 0.01 is well above what is necessary in that kind of experiment ($\mu = 10^{-10}$ was used in [12]). Depending on the kind of reference filter variations, the adaptive delay can be influenced in a more or less adverse fashion. Consider a reference filter which experiences phase and amplitude variations that are both linear. Since the variations simulated are constant across the whole frequency range, the amplitude variations correspond to a simple scaling of the reference filter impulse response. The phase variation is more problematic since it changes the shape of the impulse response. These variations incur some modifications in the quantity ξ_{\min} , which causes the delay tracking characteristics to change also. As an example, linear amplitude and phase variations were simulated, while the reference delay was kept fixed. The adaptive delay response, for a linear variation of 0.001 sample/sample, is shown in Fig. 6.24. This figure shows that the adaptive delay reacts to the variations in the reference filter. The corresponding adaptive filter impulse response, after 1000 iterations, is given in Fig. 6.25. It shows the variations in the impulse response that cause the peculiar behaviour of the adaptive delay.

6.3.5 Delay Tracking Simulations in Type II

In order to compare the behaviour of the Type I and the Type II configurations, the adaptive delay response was simulated for a reference unit delay step, when $\mu = 0.01$ and $\alpha = 0.5$, in Type II-DAB and Type II-DRB mode. The results, for noiseless conditions, are illustrated in Figs. 6.26 and 6.27. Note that the short reference impulse response of Fig. 6.1 is used. These figures should be compared to their Type I counterparts, in Figs. 6.10 and 6.11. Note first of all, that there is no overshoot in the Type II case, when $\alpha = 0.5$. The first order approximation of equation (3.24) is therefore more realistic in this case. Note also how well the adaptive delay tracks the reference delay in the Type II-DAB case, even for $\alpha = 0.1$. This last characteristic is related to the fact that the convergence speed of the adaptive filter is reduced by a delay in Type II-DAB configuration [49]. Intuitively, this fact can be explained by noting that the delay reduces the maximum gain factor μ for

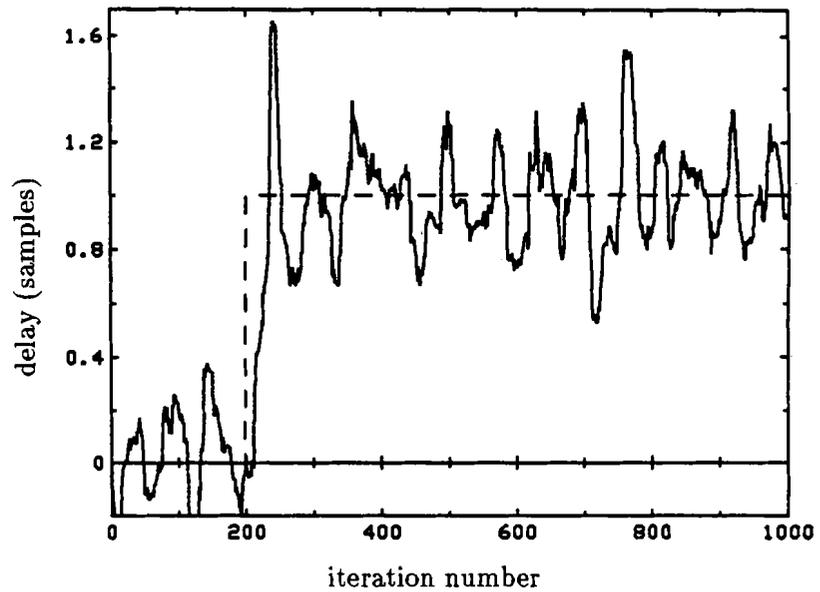


Fig. 6.21 LMS Adaptive delay response to a reference delay unit step in noisy conditions, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

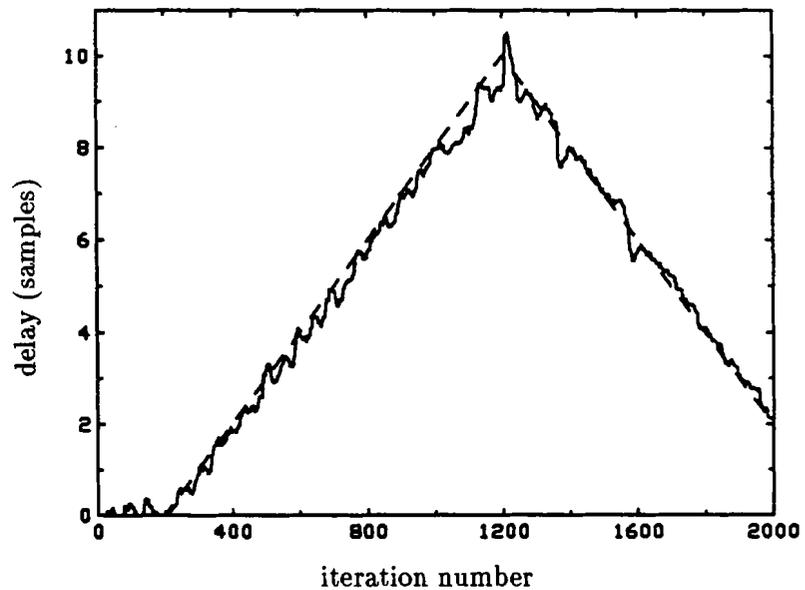


Fig. 6.22 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample in noisy conditions, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01$ and $\alpha = 0.5$

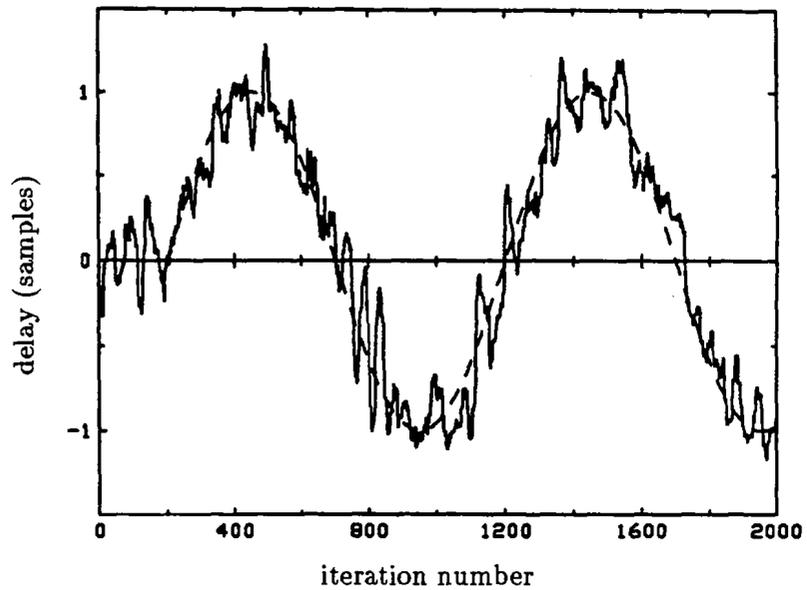


Fig. 6.23 LMS Adaptive delay response to a sinusoidal reference delay variation in noisy conditions, period = 1000 samples, amplitude = 1 sample, SNR = 10 dB; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.5$

convergence in the mean (see equation (4.145)), which itself reduces the maximum speed of convergence. Furthermore, the location of the delay after the adaptive filter “delays” the effect of any filter modifications on the error signal, which tends to slow down the speed of convergence. The time constant of delay adaptation is therefore mainly due to the adaptive delay time constant, and is similar to that of the Type I case.

In Type II-DRB configuration, this speed reduction in the adaptive filter does not exist, and the filter compensates for a portion of the delay when $\alpha = 0.1$, as in the Type I case. But note in Fig. 6.27 that there is a lag between the reference delay modification and the adaptive delay initial reaction. This is due again to the delay between the modification and its appearance in the error signal. Also, it is noticed that this lag reduces the delay convergence speed. Finally, the Type II configurations were simulated for linear and sinusoidal reference delay variations, in noiseless and noisy conditions. The results are similar to the ones for the Type I cases.

6.3.6 Discussion

The results presented in Subsections 6.3.1 to 6.3.5 establish the typical behaviour of the joint SD and LMS algorithms and make use of most of the conclusions of Chapter 3.

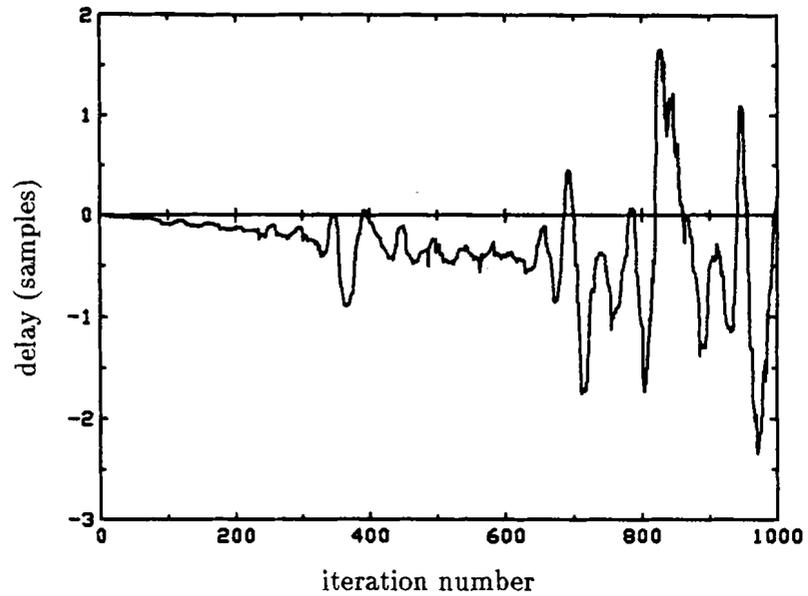


Fig. 6.24 LMS Adaptive delay response to linear phase and amplitude variations in the reference filter; variations of 0.001 sample/sample; $\mu = 0.01$, $\alpha = 0.5$

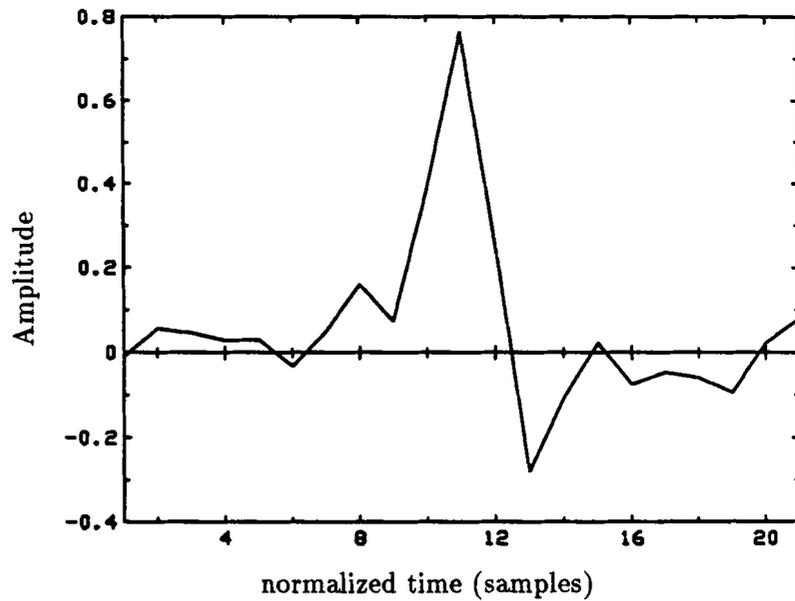


Fig. 6.25 Adaptive filter impulse response after 1000 iterations for the reference filter variations of Fig. 6.24

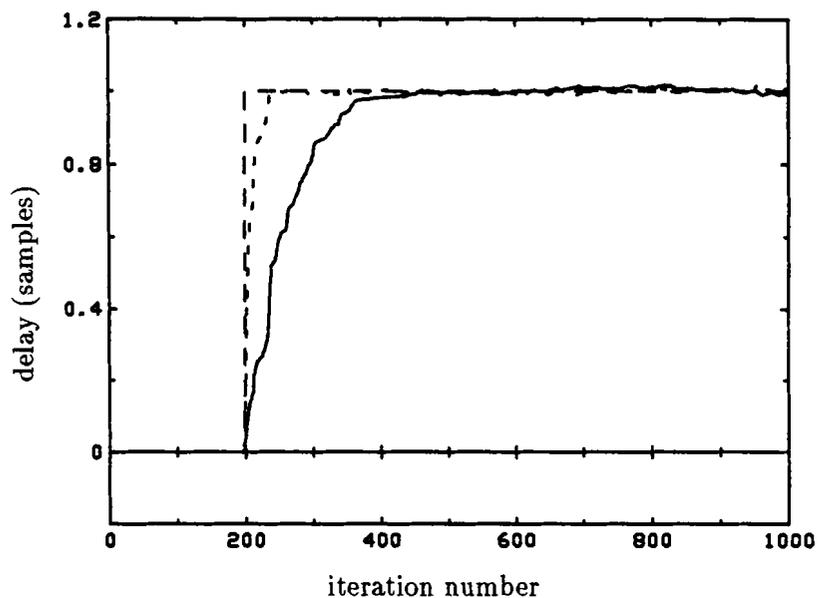


Fig. 6.26 LMS Adaptive delay response to a reference delay unit step in Type II-DAB configuration; long dashed curve: reference delay; medium dashed curve: $\mu = 0.01$, $\alpha = 0.5$; continuous curve: $\mu = 0.01$, $\alpha = 0.1$

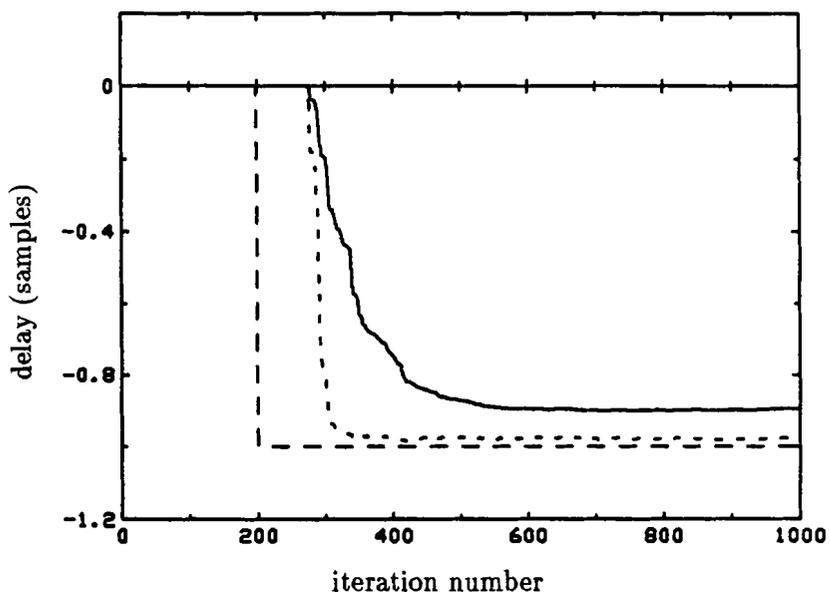


Fig. 6.27 LMS Adaptive delay response to a reference delay unit step in Type II-DRB configuration; long dashed curve: reference delay; medium dashed curve: $\mu = 0.01$, $\alpha = 0.5$; continuous curve: $\mu = 0.01$, $\alpha = 0.1$

All the relations involving $\ddot{\xi}_{\min}$ were computed using the true value of this parameter. In practice, it can be estimated by various means, one of them being the use of equation (3.117). Note however that this method can be the source of large errors. Better methods can be devised with the help of least-squares polynomial approximation or Chebyshev (min-max) polynomial approximation [51].

6.3.7 Steady-State Results

The results of Chapter 4, for the joint LMS adaptive algorithm in Type I and Type II configurations, are considered more closely in this subsection. The expected values $E[G_n^2]$ and $E[N_n^2]$, which are used in the convergence bound for α and in the steady-state delay variance v_{ss} , are first computed for a Type I and a Type II-DRB configurations. Then these quantities are used in determining α as a function of μ and v_{ss} . Finally, the excess MSE is computed for different practical cases.

6.3.7.1 Convergence Bounds and Gain Factors

The expressions for $E[G_n^2]$ and $E[N_n^2]$, for white input and noise signals, are given by equations (4.195) to (4.198). These quantities are functions of $\text{tr}[\mathbf{K}_\eta]$, which is given in equation (4.111). This equation shows that $\text{tr}[\mathbf{K}_\eta]$ is proportional to μ and v_{ss} . Since $E[G_n^2]$ and $E[N_n^2]$ are proportional to $\text{tr}[\mathbf{K}_\eta]$, these expectations are also proportional to μ and v_{ss} . For a Type I system, it is found that $E[G_n^2]$ and $E[N_n^2]$ are approximately constant for v_{ss} and μ lower than 0.01. For the Type II-DRB case, the two expectations exhibit a fairly flat response for values of v_{ss} lower than 1.0 and for values of μ lower than 0.1. This smaller sensitivity in the latter case reflects the fact that the trace operator appears only once in the Type II expectation expressions.

The expression (4.74) can be used, as suggested in the design procedure of Section 4.5, to obtain plots of α versus v_{ss} and μ . Figs. 6.28 and 6.29 show the theoretical behaviour of α as a function of μ , for both types of systems and for three different values of steady-state variance. The gain factor α increases with v_{ss} and for a typical variance of 0.01, the value of α is approximately constant with μ , and is around 0.5. This indicates that, for low variance, the adaptive filter does not influence much the noisy behaviour of the adaptive delay. The upper bound on α for convergence in the mean square (equation (4.73)) is illustrated in Figs. 6.30 and 6.31 for the same conditions. The delay variance does not influence much this upper bound, which is approximately constant for $\mu < 0.01$.

The theoretical behaviour of α as a function of v_{ss} , and for two different signal-to-noise ratios, is illustrated in Figs. 6.32 to 6.35. The gain factor α is seen to be proportional to the

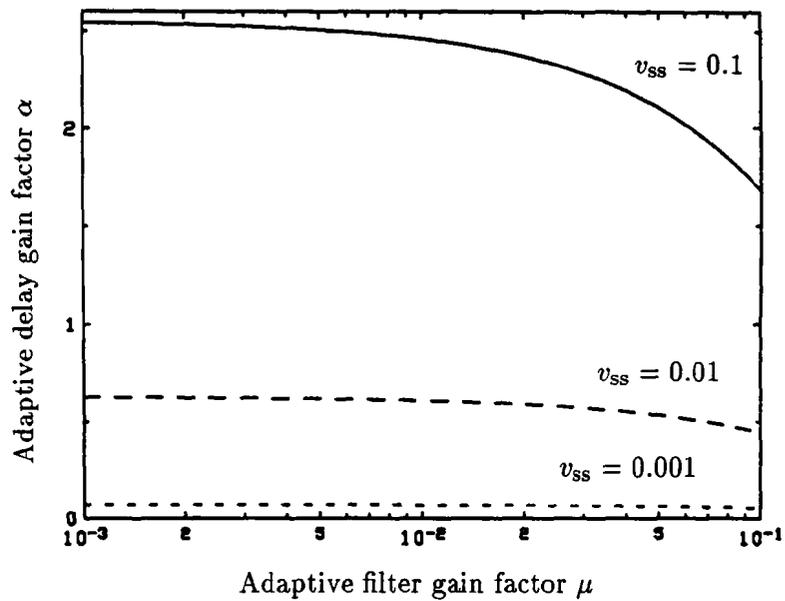


Fig. 6.28 Theoretical curve of α versus μ for a Type I system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$

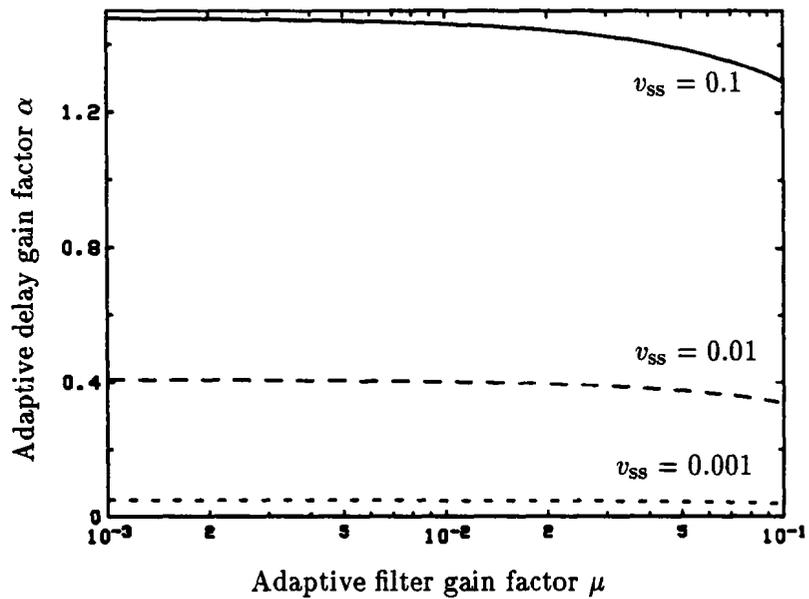


Fig. 6.29 Theoretical curve of α versus μ for a Type II-DRB system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$

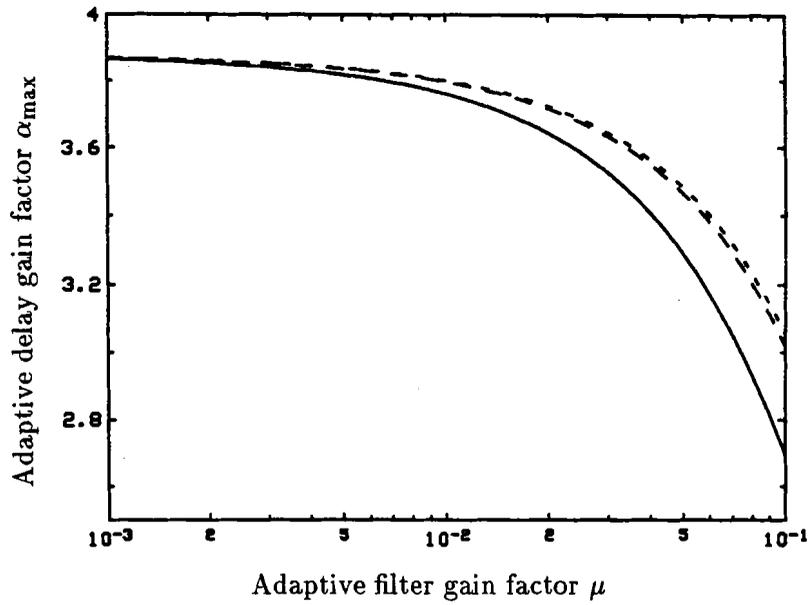


Fig. 6.30 Theoretical curve of α_{\max} versus μ for a Type I system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$

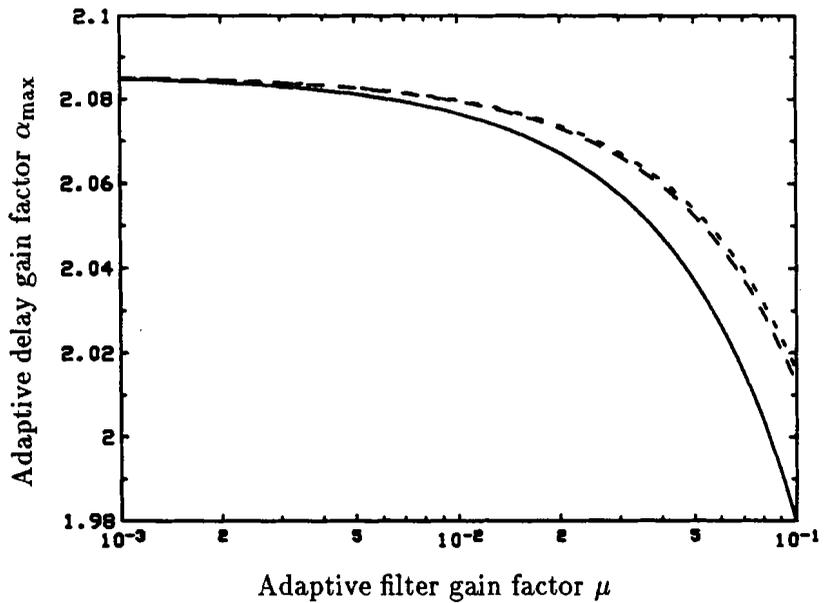


Fig. 6.31 Theoretical curve of α_{\max} versus μ for a Type II-DRB system; SNR = 10 dB; small dashes curve: $v_{ss} = 0.001$, large dashes curve: $v_{ss} = 0.01$, continuous curve: $v_{ss} = 0.1$

variance for lower values of v_{ss} . For higher values of v_{ss} , α is limited by the upper bound for mean square convergence.

The design procedure of Section 4.5 is based on plots similar to those of Figs. 6.28 and 6.29. In this particular case, these plots show that, for a given variance, μ can be chosen over a large range without affecting the behaviour of the delay estimation. This fact was already noticed in the simulations.

6.3.7.2 Excess Mean Squared Error

A major result from Chapter 4 is the expression for the excess MSE at the output of the joint LMS algorithm. For all types of joint algorithms, the expression is of the form

$$\xi_{ex} = \xi_{ex}^d + \xi_{ex}^f + \xi_{ex}^{df},$$

or, in term of misadjustments,

$$\mathcal{M} = \mathcal{M}^d + \mathcal{M}^f + \mathcal{M}^d \mathcal{M}^f.$$

These results are verified for a Type I system by computing the theoretical value of ξ_{ex}^f , using equation (4.123), and by obtaining ξ_{ex}^d as well as ξ_{ex} by simulations. The results, for five different combinations of α and μ , are presented in Table 6.2. The corresponding measured total misadjustment \mathcal{M} is obtained from ξ_{ex} by dividing by ξ_{min} , while the theoretical total misadjustment \mathcal{M}_{th} is obtained using equation (4.125). This table shows the good agreement between the measured and the theoretical quantities. Note that the cross-product term $\mathcal{M}^d \mathcal{M}^f$ being a second order component, its effect is therefore small or negligible, as can be seen from the fact that ξ_{ex} is always approximately equal to the sum of ξ_{ex}^f and ξ_{ex}^d .

μ	α	ξ_{ex}^f	ξ_{ex}^d	ξ_{ex}	\mathcal{M}	\mathcal{M}_{th}
0.1	0.5	0.00312	0.00193	0.00563	40.5%	39.4%
0.05	0.5	0.00141	0.00193	0.00308	22.1%	25.4%
0.1	0.1	0.00312	0.00010	0.00313	22.5%	23.3%
0.01	0.5	0.00026	0.00193	0.00195	14.0%	16.0%
0.05	0.25	0.00141	0.00051	0.00163	11.7%	14.2%

Table 6.2 Excess mean squared errors and misadjustments for different combinations of α 's and μ 's

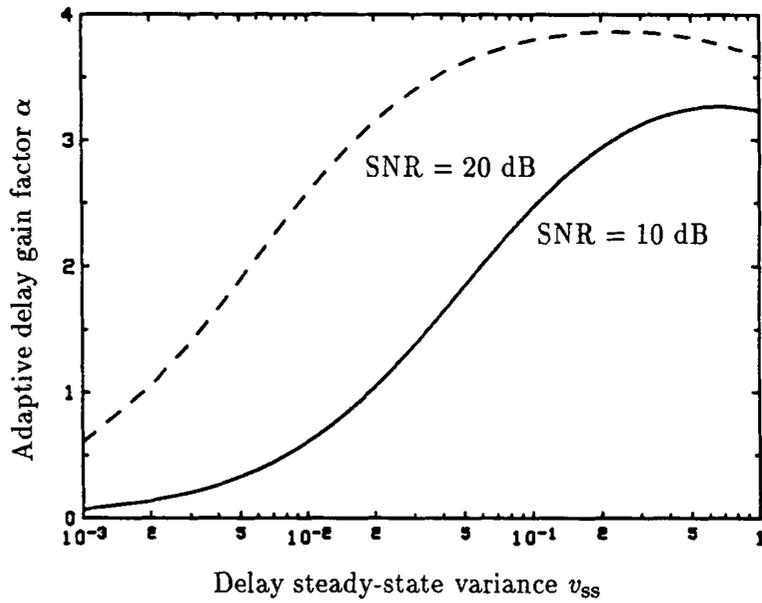


Fig. 6.32 Theoretical curve of α versus v_{ss} for a Type I system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB

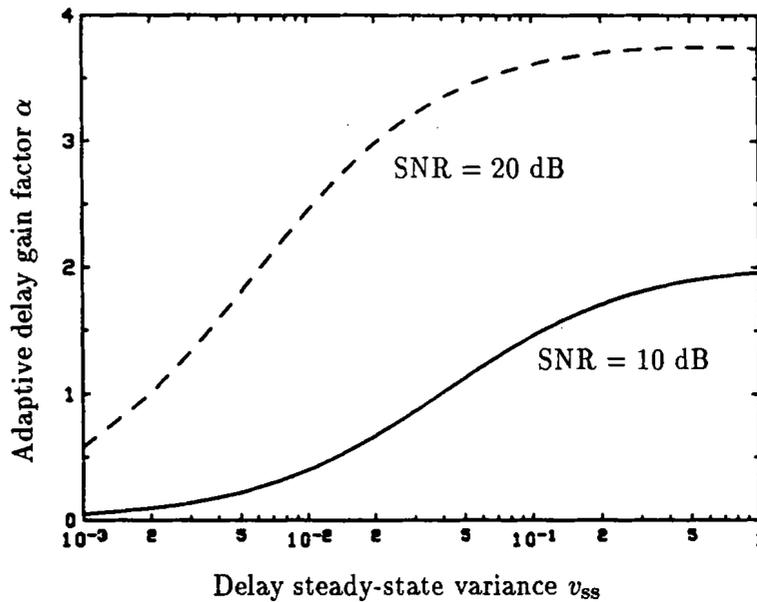


Fig. 6.33 Theoretical curve of α versus v_{ss} for a Type II-DRB system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB

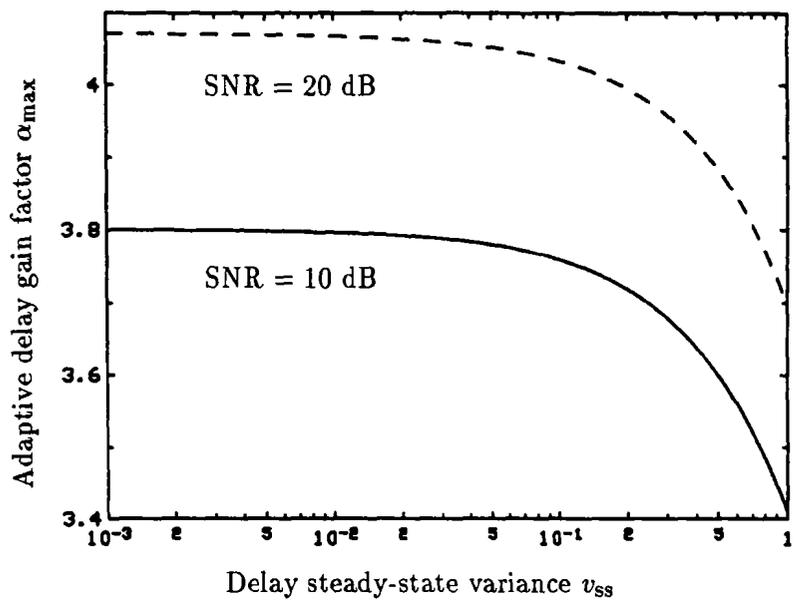


Fig. 6.34 Theoretical curve of α_{\max} versus v_{ss} for a Type I system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB

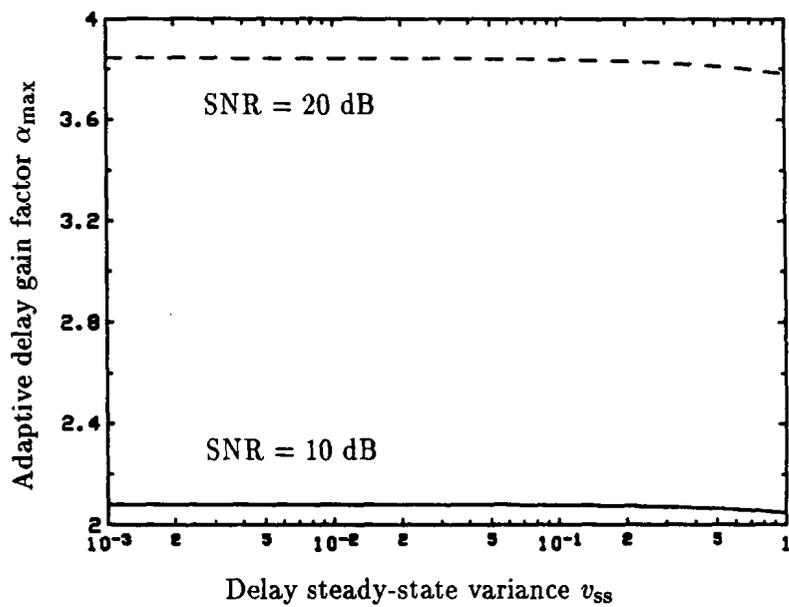


Fig. 6.35 Theoretical curve of α_{\max} versus v_{ss} for a Type II-DRB system; $\mu = 0.01$; continuous curve: SNR = 10 dB, dashed curve: SNR = 20 dB

The total misadjustment, for a Type I system, is illustrated in Fig. 6.36 as a function of the steady-state delay D , for an adaptive filter operating alone and for a joint adaptive system. The misadjustment for the latter system is essentially constant with respect to the delay, while it is a function of D in the former case. This figure shows that for a delay lower than 9 samples, the adaptive filter alone produces a smaller relative error, but for larger delays, the misadjustment due to the coupled adaptive processes is inferior to the misadjustment produced by the single filter.

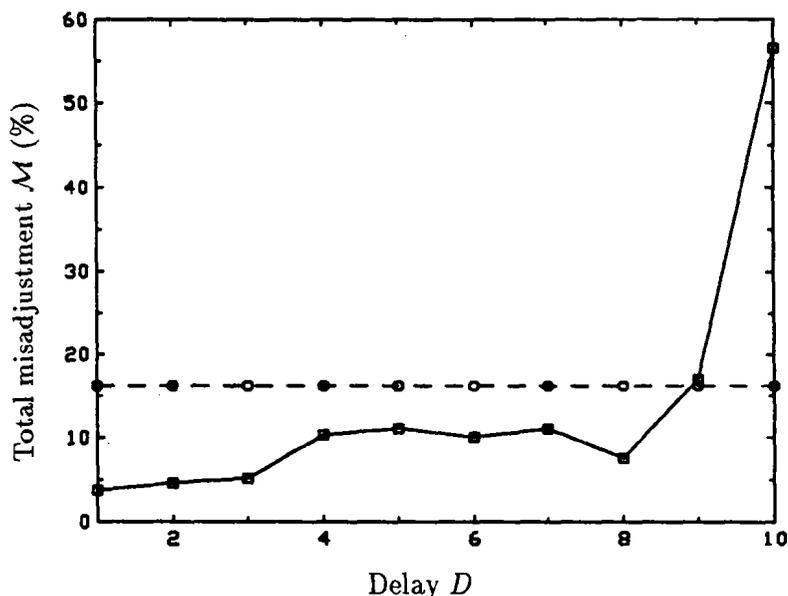


Fig. 6.36 Measured misadjustment for a Type I system versus the steady-state delay D , SNR = 10 dB, $\mu = 0.01$, $\alpha = 0.5$; continuous curve: adaptive filter alone, dashed curve: joint adaptive system

It was noted theoretically in Chapter 4 that, in a Type II-DAB system, the excess MSE is increased by the presence of the adaptive delay after the filter (see equations (4.177) and (4.178)). This result is confirmed in practice in Fig. 6.37 where the total measured excess MSE is illustrated as a function of the steady-state delay D .

6.4 Results with the Joint RLS Algorithm in Type II-DRB Configuration

The behaviour of the sum of squared errors $\hat{\xi}_{M_0}(n, d)$ with respect to d and β is first investigated in this section. The numerical stability of the algorithm is discussed in Subsec-

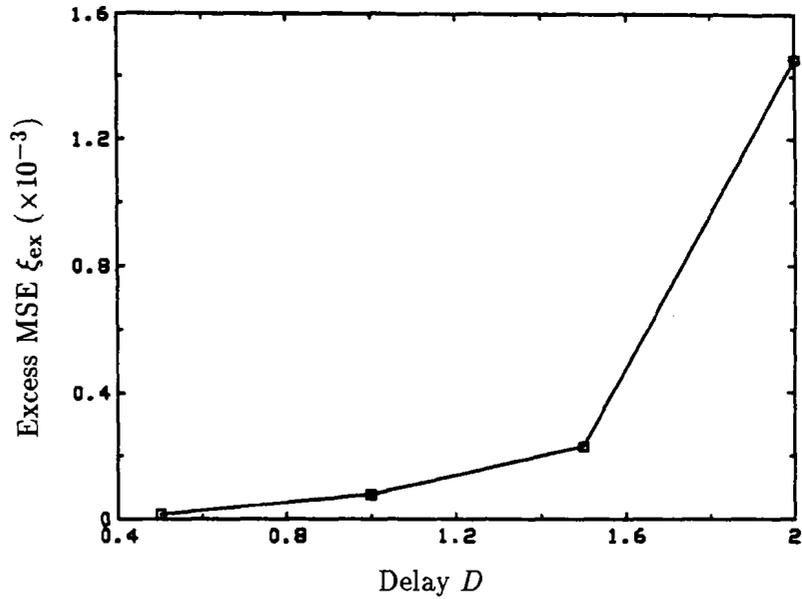


Fig. 6.37 Measured excess MSE for a Type II-DAB system versus the steady-state delay D , $\mu = 0.01$, $\alpha = 0.5$

tion 6.4.2. The tracking properties of the algorithm are then considered in Subsection 6.4.3, where the simulation results are given for different channel characteristics.

The only configuration simulated with the joint RLS algorithm was the Type II-DRB one. The algorithm of Subsection 5.4.1 was essentially implemented integrally, except for an extra set of computations used to stabilize it numerically.

6.4.1 The Sum of Squared Errors

In order to verify the behaviour of the sum of squared errors, when there is a nonzero relative delay Δ between the reference delay D and the adaptive integer delay ℓ , the sum of squared errors is first obtained as a function of Δ and is illustrated in Figs. 6.38 and 6.39. Note that the adaptive system is in steady-state prior to time $n = 0$ and that the delay difference is applied at $n = 0$.

It is noticed that after a transient period of approximately 200 iterations, $\hat{\xi}_{M_o}(n, \ell)$ takes an average value that increases with the absolute value of Δ . Note also that the randomness in $\hat{\xi}_{M_o}(n, \ell)$ is due to the input signals stochastic behaviour. The steady-state expected value of $\hat{\xi}_{M_o}(n, d)$ versus $\Delta = D - d$ is given in Fig. 6.40. Note that the oscillatory behaviour of $E[\hat{\xi}_{M_o}(n, d)]$ is due to the oscillations in the reference filter and in the input signal autocorrelation (see the expressions for the MSE functions in equations (3.64) to (3.66)). Note also that in this particular case, as long as the relative delay is smaller than

2 samples, a delay adaptation based on $\hat{\xi}_{M_o}(n, \ell - 1)$, $\hat{\xi}_{M_o}(n, \ell)$ and $\hat{\xi}_{M_o}(n, \ell + 1)$ has the potential to bring the relative delay to zero. But for a larger initial relative delay, it is also possible that, because of the oscillations in $E[\hat{\xi}_{M_o}(n, d)]$, the delay adaptation algorithm locks on a false value.

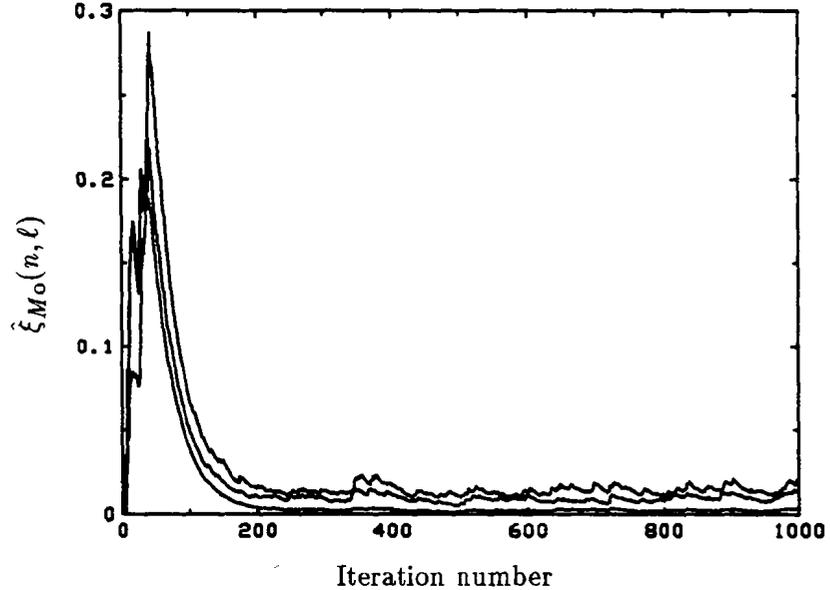


Fig. 6.38 Minimum sum of squared errors versus time, for different relative delays Δ and for $\beta = 0.97$; the lowest curve is for $\Delta = -1$, the middle curve is for $\Delta = 2$ and the upper one if for $\Delta = 6$.

Another interesting characteristic of $\hat{\xi}_{M_o}(n, \ell)$ is its behaviour with respect to β . From equations (5.1) and (5.4), it is seen that the memory of the algorithm is proportional to β . This implies that when the forgetting factor increases, the number of significant terms in $\hat{\xi}_{M_o}(n, \ell)$ also increases, causing the value of the sum to grow. This is illustrated in Figs. 6.41 and 6.42 for three values of β . The measured expected value and variance of $\hat{\xi}_{M_o}(n, \ell)$, in steady-state and for a relative delay of two samples, are shown in Figs. 6.43 and 6.44.

6.4.2 Numerical Stability

It is well known that the FTF implementation of the RLS algorithms is inherently unstable, when a finite word length machine and a forgetting factor β lower than one are used [68]. This phenomenon is due to the instability of the system through which the finite precision error is propagated. Since the introduction of the different forms of the fast RLS algorithms, several methods were proposed to stabilize their behaviour.

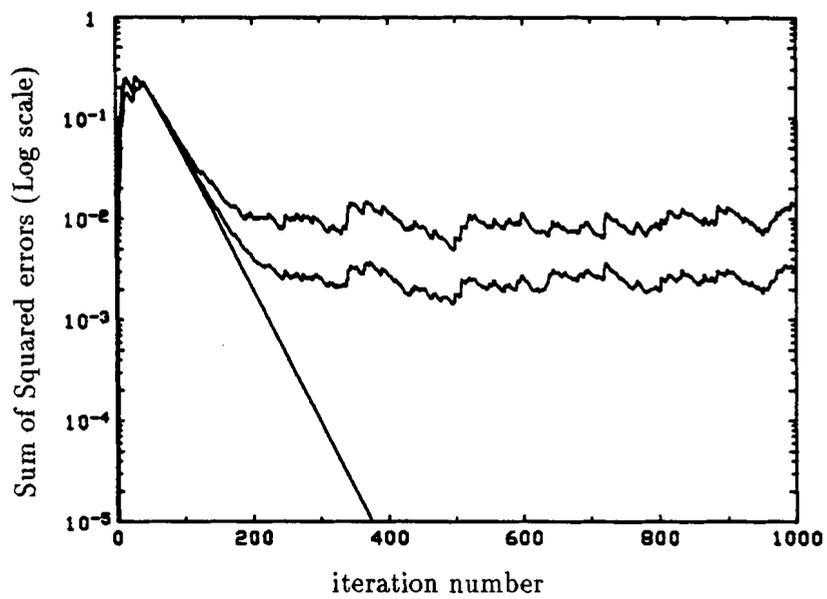


Fig. 6.39 Fig. 6.38 on a vertical log scale

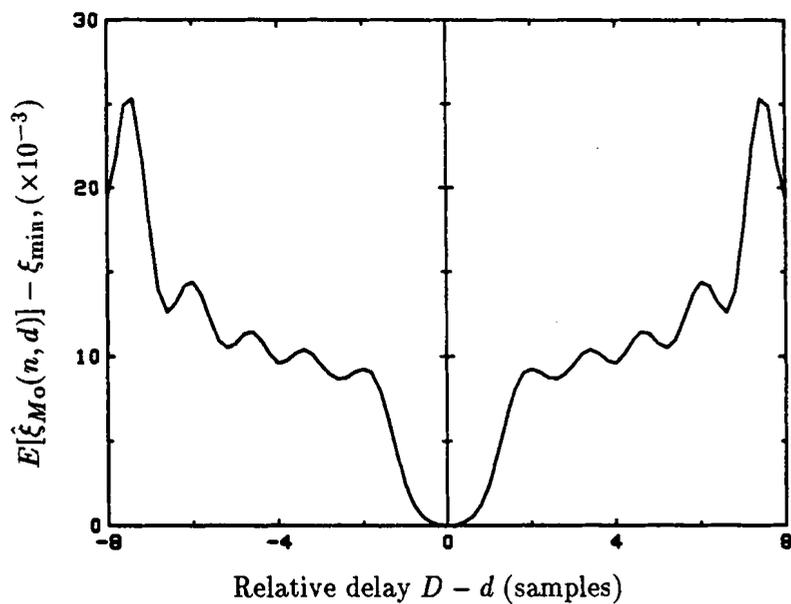


Fig. 6.40 Minimum sum of squared errors versus $D - d$, $\beta = 0.97$.

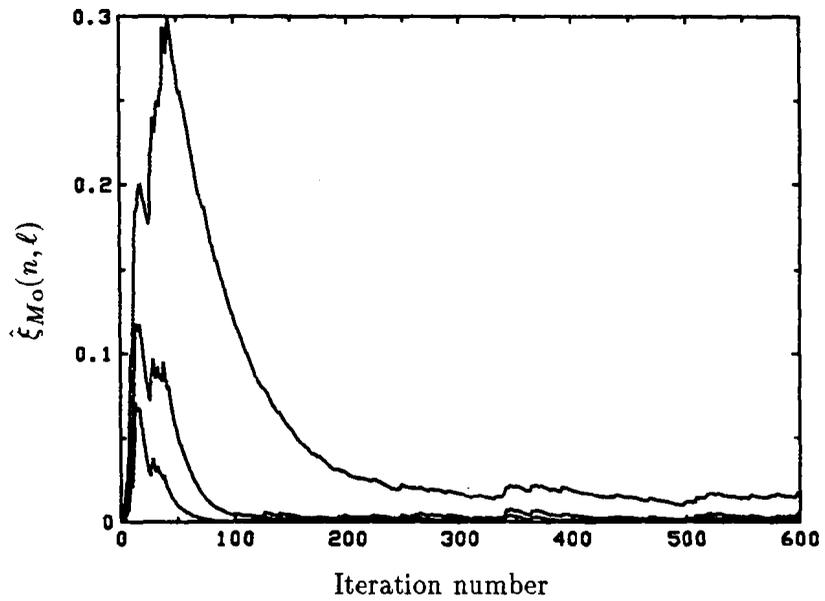


Fig. 6.41 Minimum sum of squared errors versus time, for different values of β and for a relative delay of two sample; the lowest curve is for $\beta = 0.9$, the middle is for $\beta = 0.94$ and the upper one is for $\beta = 0.98$.

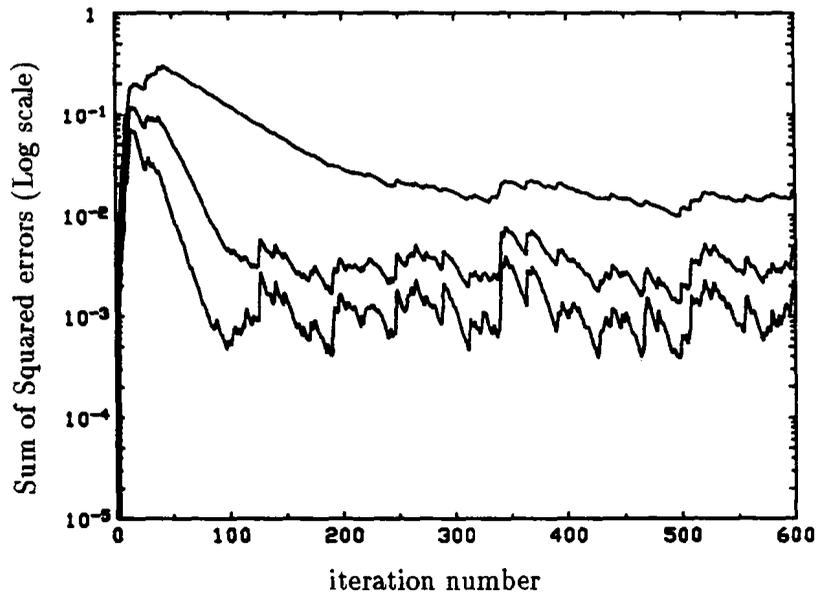


Fig. 6.42 Fig. 6.41 on a vertical log scale

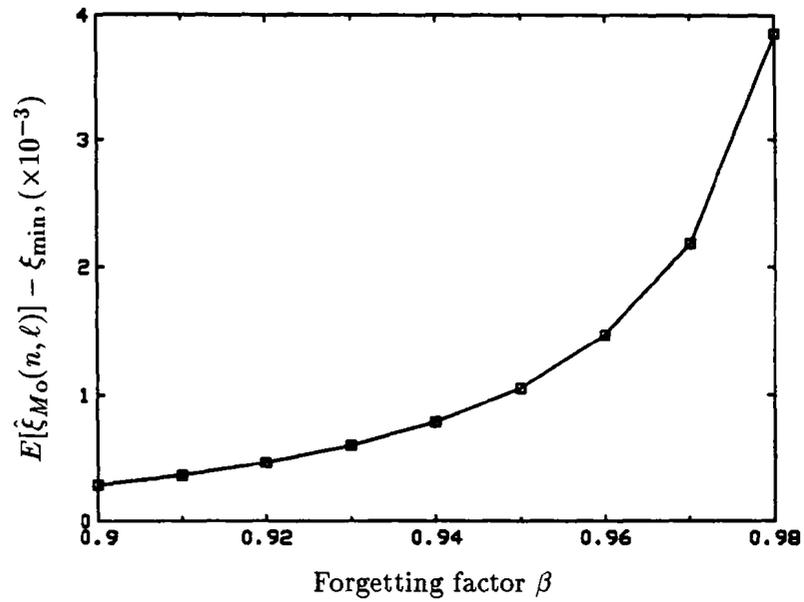


Fig. 6.43 Measured expected value of the minimum sum of squared errors versus β .

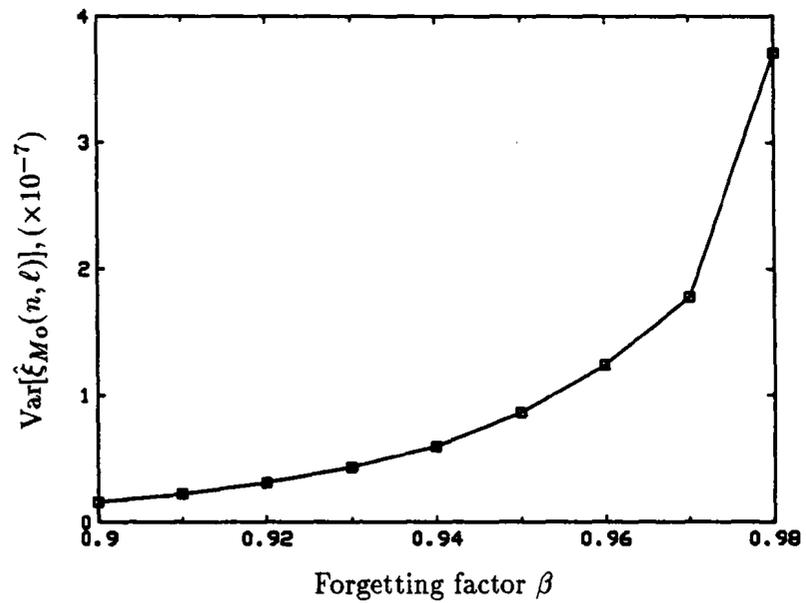


Fig. 6.44 Measured variance of the minimum sum of squared errors versus β .

Lin suggested the monitoring of a specific variable, in the fast algorithm, which is shown by simulations to become negative when the algorithm diverges [69]. This *rescue device* is therefore used to decide upon when the algorithm should be reinitialized, such that the finite precision error accumulation is zeroed. Eleftheriou and Falconer used a periodic restart procedure in which the fast algorithm is interrupted and restarted at periodic intervals, with a parallel LMS algorithm taking over for the reinitialization period [33]. More recently, some researchers proposed more fundamental modifications to the algorithm such that the error propagation mechanism is directly stabilized. Slock and Kailath introduced redundancy in the algorithm, which allows the feedback of numerical errors and the “correction” of such errors in a channel coding manner [70]. Benallal and Gilloire applied some control principles to the linear system governing the error propagation, such that the system is stabilized without changing the theoretical form of the overall algorithm [71].

The focus of the present research being on the joint delay estimation and adaptive filtering capabilities of the algorithms, it was felt that only a rather crude stabilization mechanism was necessary in the simulations. Therefore, a periodic restart procedure was introduced, in which a parallel version of the FTF algorithm was periodically started, and its resulting parameters transferred to the main FTF algorithm after a number of iterations large enough to ensure convergence. This parallel periodic restart procedure is reminiscent to the method used by Eleftheriou and Falconer, although more computationally involved. It was felt that this method would interfere the least into the other aspects of the joint algorithm.

In the simulations performed, it was noticed that the joint algorithm becomes unstable after 600 to 700 iterations, especially for lower values of β . The restart period was therefore fixed to 500 iterations for most of the simulations. The parallel algorithm begins 200 iterations before the transfer of the newly computed intermediate variables.

The resulting behaviour of the sums of squared errors is illustrated in Fig. 6.45, where $\hat{\xi}_{M_o}(n, \ell - 1)$ is plotted for 3000 iterations and $\beta = 0.92$. The algorithm is therefore seen to be stabilized by the parallel restart procedure. The behaviour of the error in the parallel implementation is illustrated in Fig. 6.46, where the sum of squared errors is seen to experience a sudden increase every 500 iterations and settles down well within the 200 iterations period allocated before the transfer of information to the main algorithm. These two figures illustrate that the stabilization procedure performs as expected and that the simulation results obtained in the next section are illustrative of the potential of the joint algorithm.

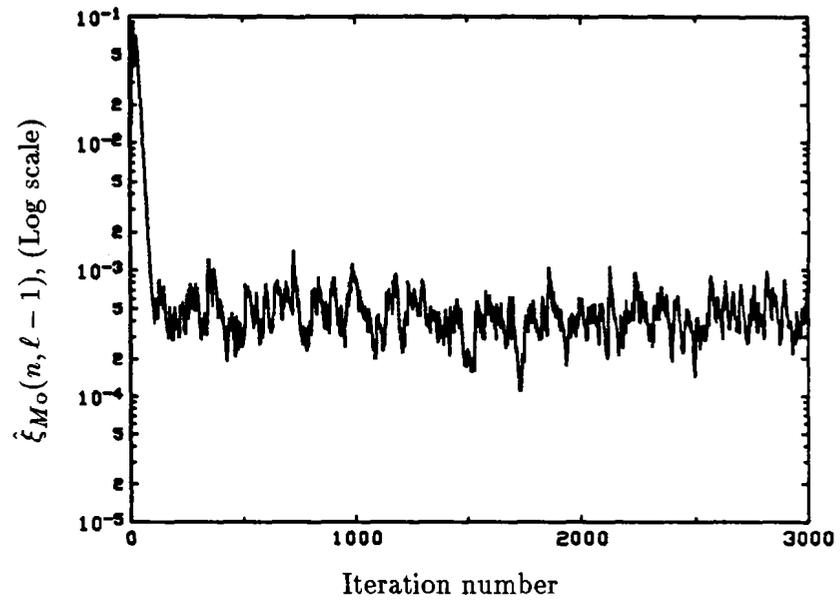


Fig. 6.45 Behaviour of $\hat{\xi}_{M_0}(n, \ell - 1)$ with parallel restart every 500 iterations, $\beta = 0.92$.

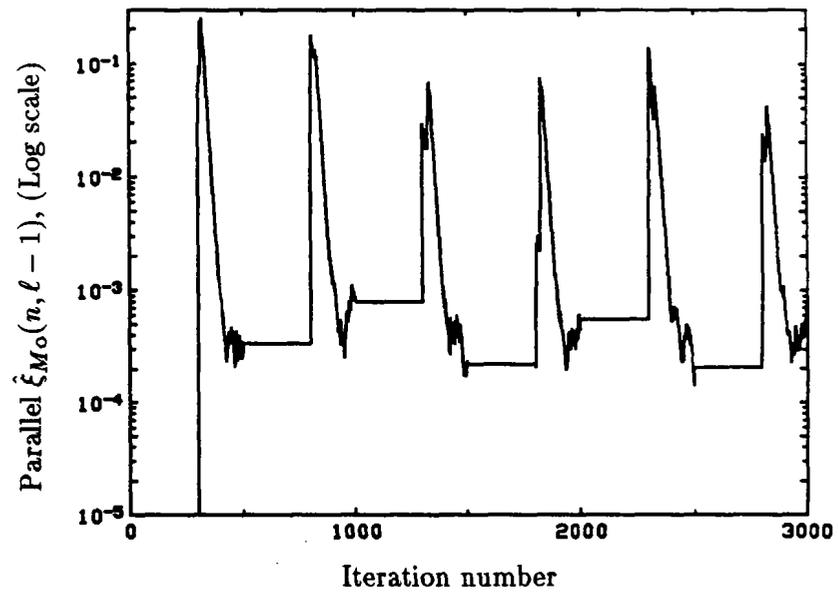


Fig. 6.46 Behaviour of the parallel $\hat{\xi}_{M_0}(n, \ell - 1)$ with parallel restart every 500 iterations, $\beta = 0.92$.

6.4.3 Tracking Properties

The tracking properties of the joint RLS algorithm are simulated in this section. In order to perform the lag-update decision (Part c of Subsection 5.4.1) the time average of the sum of squared errors must be computed. This is done by accumulating the sum of squared errors over 50 iterations.

The adaptive delay responses to a linearly changing reference delay are presented in Figs. 6.47 to 6.49. The reference slope is 0.01 sample/sample, as for the joint LMS algorithm case. The noiseless case is shown in Fig. 6.47, and the results for SNR's of 30 dB and 20 dB appear in the two other figures. Except for a granular-type of noise, the adaptive delay tracks well the reference delay. Note that the forgetting factor β was set to 0.92, in order to allow good tracking. The results for a sinusoidal reference delay are illustrated in Figs. 6.50 to 6.52. Adequate tracking is again demonstrated in this case.

6.4.4 Discussion

The simulations of the joint RLS algorithm presented in this section indicate that the development of Chapter 5 leads to a potentially very useful algorithm. By averaging the minimum sums of errors over 50 samples, and by comparing three of these sums of errors, the delay tracking is very good in all cases for SNR's as low as 20 dB. Below this value, the performances degrade very quickly. But for each application, there is an optimum strategy for delay estimation, and the particular one chosen here is fairly empirical. This simple method shows that the joint RLS algorithm can keep the adaptive filter impulse response approximately centered in many different kinds of scenarios. It indicates also that if rapid adaptation to the reference filter is required and that computational complexity is a secondary issue, the conventional RLS adaptive filter can be favorably enhanced by the delay estimation based on the lag-recursive relations.

6.5 Results for a Reverberant Room Reference Impulse Response

In order to test the joint LMS algorithm in a more practical context, an impulse response typical of a reverberant room is used in the reference filter. This response is 200-tap long and is generated using the method proposed by Allen and Berkley [72]. It simulates the behaviour of a 6 metres by 6 metres room with a height of 3 metres. The reflection coefficient of the walls is 0.8, the sound source is assumed located about 0.5 metre away from one of the corners and the location of the receiver is about one metre from the same corner.

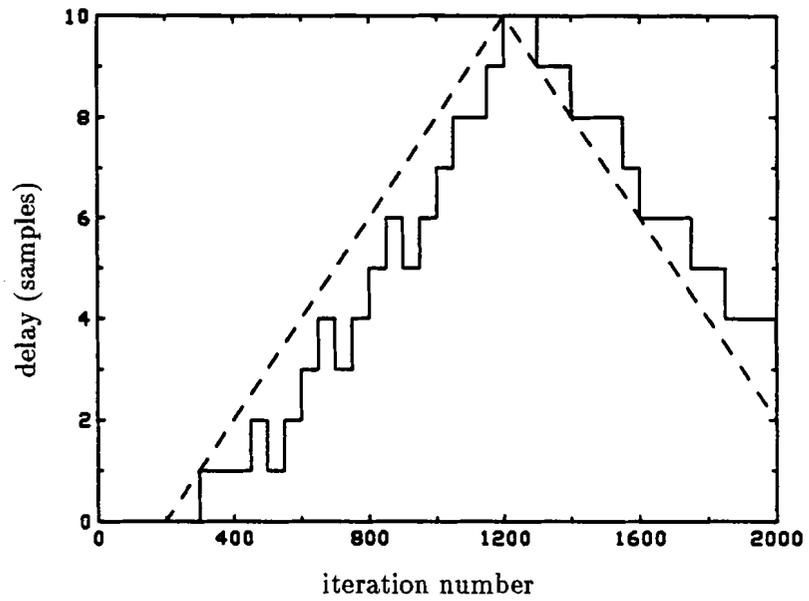


Fig. 6.47 Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, noiseless conditions

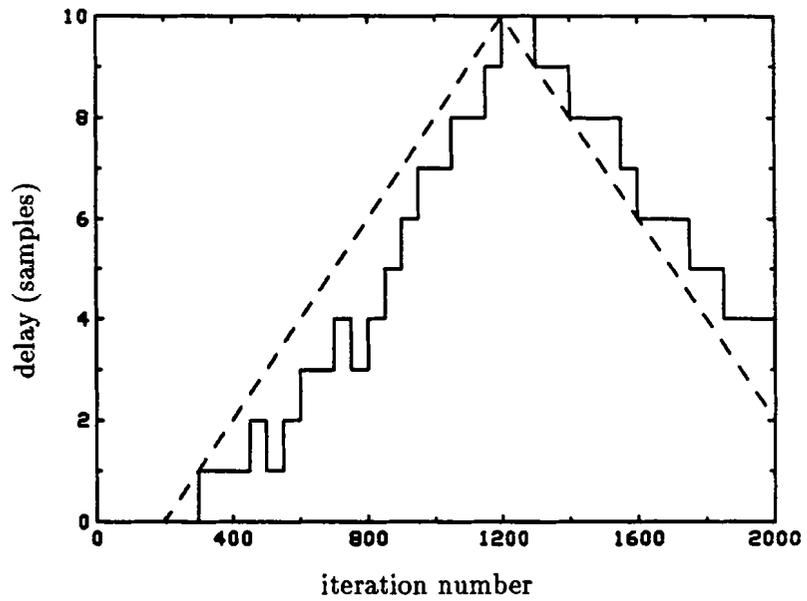


Fig. 6.48 Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 30 dB

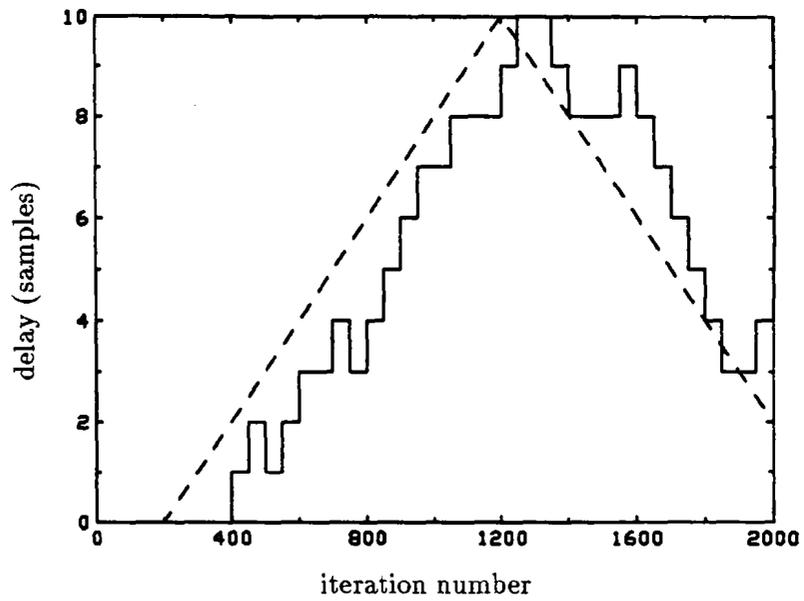


Fig. 6.49 Tracking of a linearly changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 20 dB

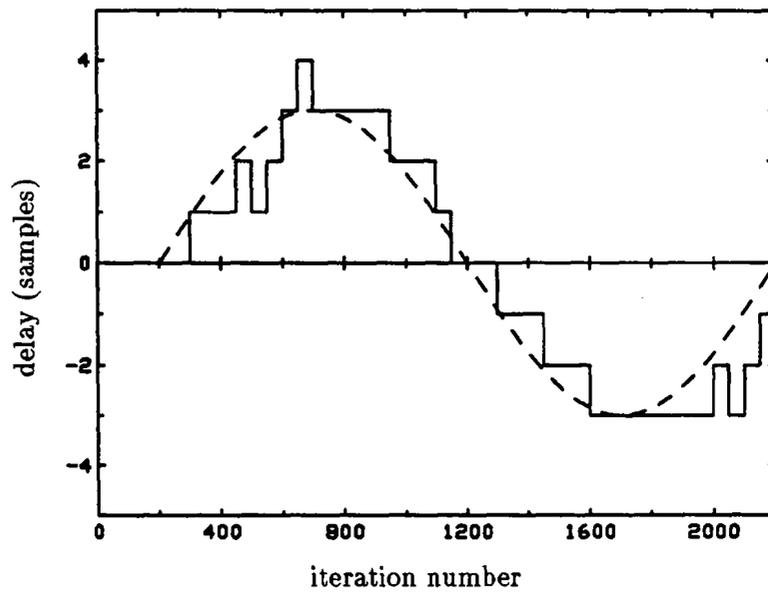


Fig. 6.50 Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, noiseless conditions

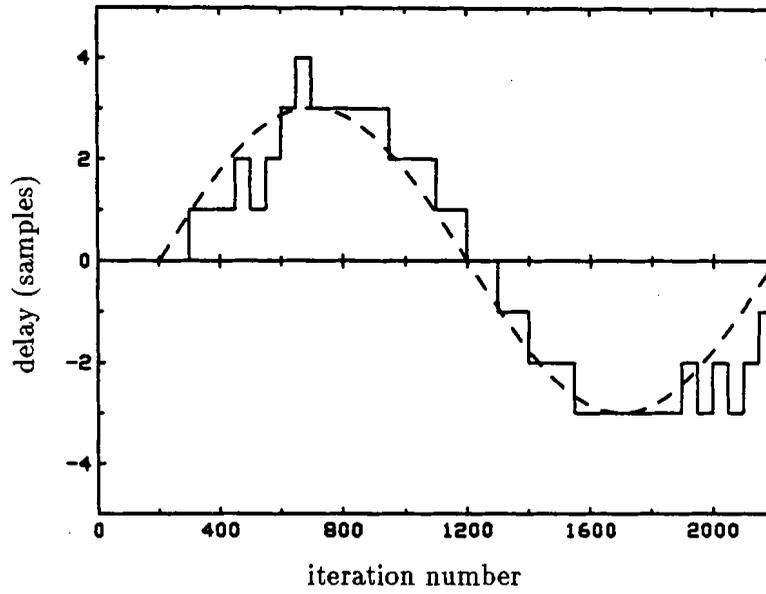


Fig. 6.51 Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 30 dB

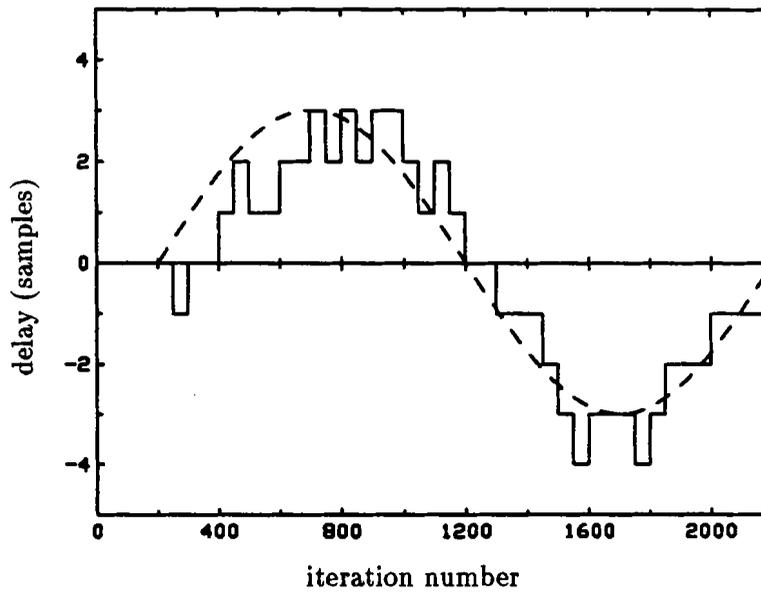


Fig. 6.52 Tracking of a sinusoidally changing delay; dashed line: reference delay, continuous line: adaptive delay, $\beta = 0.92$, SNR = 20 dB

The corresponding impulse response is given in Fig. 6.53. Note that the response is not symmetrical with respect to any point, as is the 21-tap response of Fig. 6.1, and that it exhibits three large reflection peaks as well as five smaller ones.

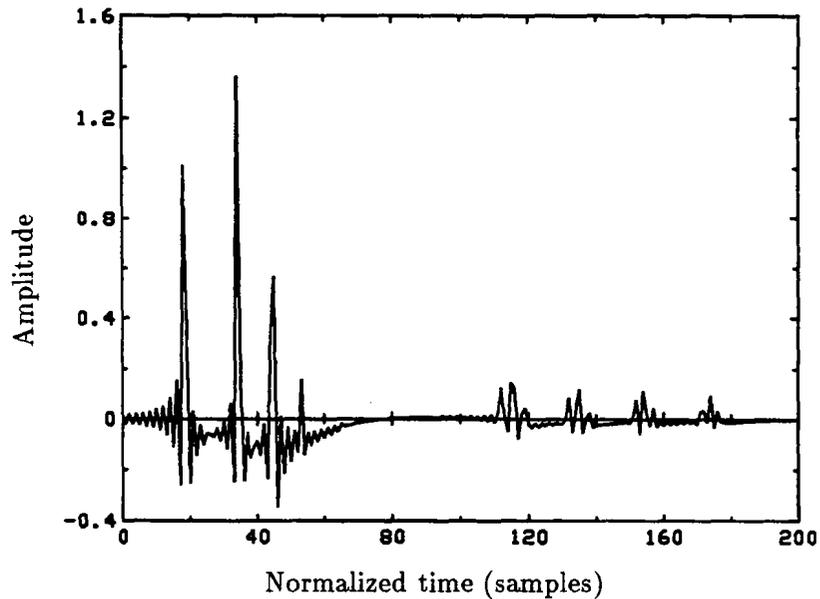


Fig. 6.53 Impulse response of the reverberant room

6.5.1 Results with the Joint LMS Algorithm in Type I

The joint LMS algorithm, with a 200-tap adaptive filter, is first simulated with a white and a coloured input Gaussian signal, in noiseless conditions. Then a digitized speech segment input is used with a normalized form of the adaptive delay algorithm. For the Gaussian input case, it is noted that the adaptive filter gain factor μ has to be lower than that for the short impulse response, otherwise the algorithm is unstable. This is predicted in Proposition 4.5, which states that, for convergence in the mean square, μ must be lower than the inverse of the trace of the input signal autocorrelation matrix. With an adaptive filter that has an order of magnitude more coefficients, it is expected that the maximum on μ be consequently smaller. In practice, it is found this maximum must be around 0.01. This value is used in the simulations, which prevents the adaptive filter from tracking fast channel variations, in particular fast reference delay nonstationarities.

6.5.1.1 White Gaussian Input

The delay tracking of the joint algorithm is shown in Figs. 6.54 and 6.55, for a reference delay ramp and a sinusoidal reference delay in noiseless conditions.

The delay tracking is seen to be good. Note the different behaviour of positive and negative delay tracking, especially in Fig. 6.55. This difference is related to the fact that the reference impulse response is not symmetrical with respect to any of its points. In order to appreciate the effectiveness of the joint algorithm, the learning curve corresponding to the joint algorithm facing a linearly changing delay (corresponding to Fig. 6.54) is illustrated in Fig. 6.56, and the learning curve corresponding to the adaptive filter coping *alone* with the same linear reference delay is illustrated in Fig. 6.57. As before, these curves were obtained by averaging 10 error curves. Note the scale difference between Fig. 6.56 and Fig. 6.57. It is obvious from these figures that the joint algorithm generates a MSE lower than the MSE for the single adaptive filter. This is also the case for a sinusoidal reference delay, as it is illustrated in Figs. 6.58 and 6.59. Note that there is a factor of 10 between the vertical scales of these two figures.

It is also interesting to compare the adaptive filter impulse response, in the joint algorithm, to the reference one. The former one is illustrated in Fig. 6.60 for the case of a reference delay ramp in noiseless conditions and after 1000 iterations. Note the algorithm error that is superimposed on the reference filter estimate. This error is responsible for a portion of the steady-state MSE generated by the algorithm.

6.5.1.2 Coloured Gaussian Input

In order to generate a coloured Gaussian input, a white Gaussian signal is passed through a filter with a non-flat transfer function. The selected frequency response is illustrated in Fig. 6.61. It exhibits in-band amplitude variations on the order of 10 dB.

The delay tracking, by the joint algorithm, of a reference delay ramp and a sinusoidal reference delay is illustrated in Figs. 6.62 and 6.63 in noiseless conditions. The adaptive delay is again seen to be adequate.

6.5.1.3 Speech Input

The segment of digitized speech used for the experimentations is illustrated in Fig. 6.64. It is part of a speech data file sampled at 8 kHz. This segment was selected such that a large range of amplitude variations is present over its span. The dashed line indicates the range of data used for initializing and training the different algorithms, and the range actually

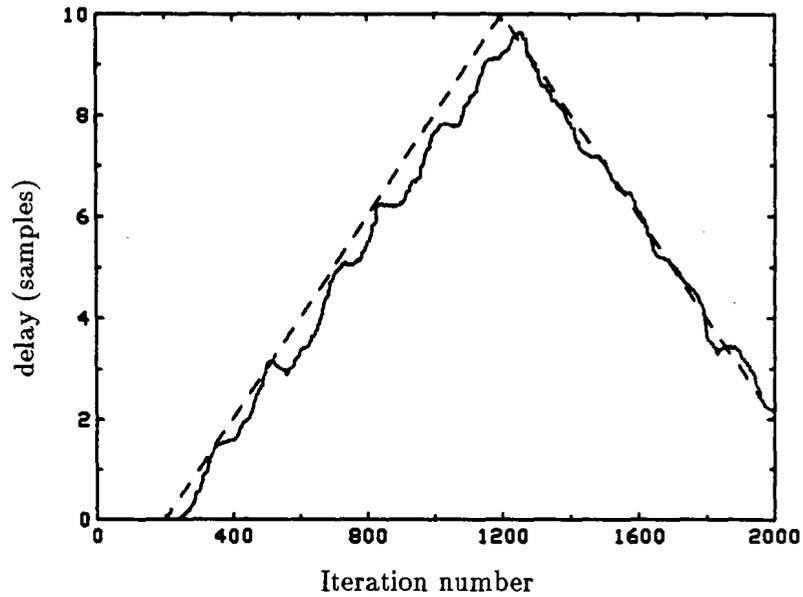


Fig. 6.54 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample and for a 200-tap reference impulse response; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$

used for delay tracking. The data up to the dashed line is used for training and the rest is used for tracking.

Because of larger input data spectral variations, which translate into a larger eigenvalue spread, the adaptive filter gain factor has to be lowered. A value of $\mu = 10^{-5}$ is used. The input signal variations prevent the adaptive delay algorithm to perform properly when the input amplitude decreases too much. The algorithm of equation (4.33) is therefore modified into the *normalized* form

$$d_{n+1} = d_n + \frac{2\alpha\epsilon(n, d_n)\mathbf{w}_n^T \dot{\mathbf{u}}(nT - d_n)}{\|\mathbf{u}_n\|^4}, \quad (6.24)$$

where the square of the input power is defined as

$$\|\mathbf{u}_n\|^4 = \left(\sum_{i=0}^{199} u^2(n-i) \right)^2. \quad (6.25)$$

A fourth power is needed for amplitude normalization, since the error and the input vector are each proportional to the amplitude, while the weight vector is proportional to its square (see equation (4.32)).

Once normalized, the adaptive delay can track more adequately the reference delay variations, even when the amplitude is reduced, as it is the case around the 2500th iteration on Fig. 6.64. Note however that the adaptive delay gain factor α has to be increased by four

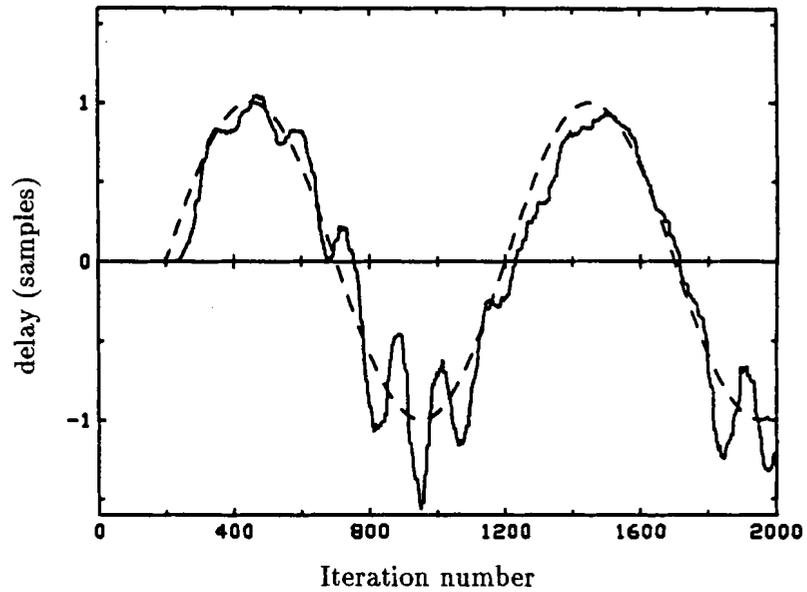


Fig. 6.55 LMS Adaptive delay response to a sinusoidal reference delay variation and for a 200-tap reference impulse response; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$

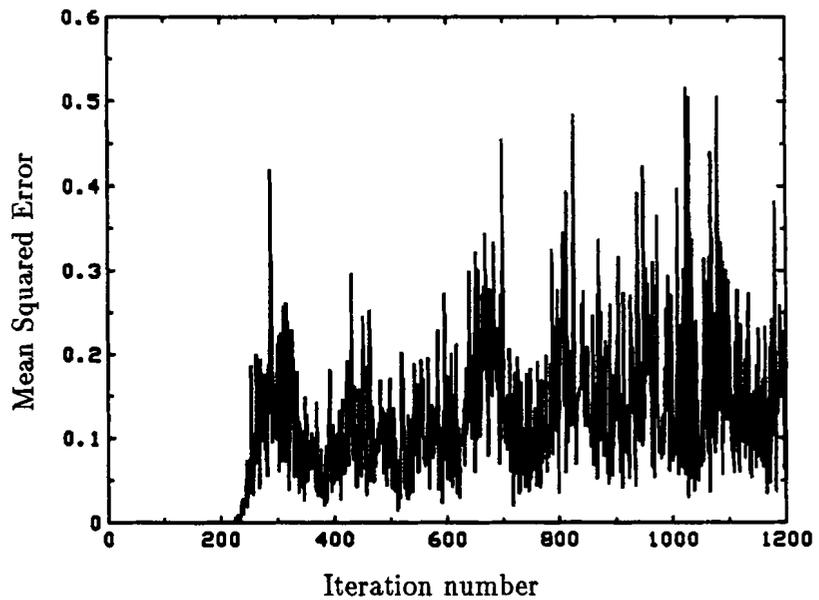


Fig. 6.56 Learning curve for the joint algorithm facing a reference delay ramp of 0.01 sample/sample (corresponding to Fig. 6.54); $\mu = 0.01$, $\alpha = 0.02$

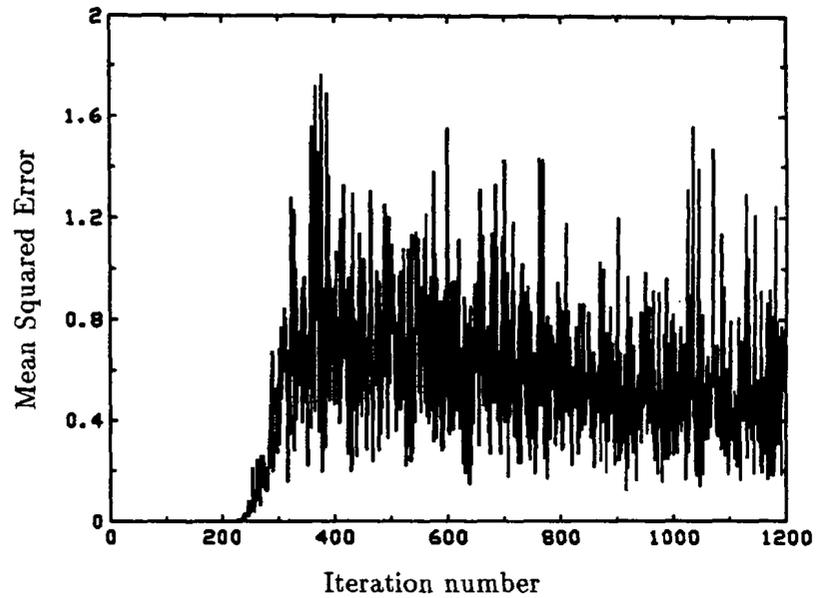


Fig. 6.57 Learning curve for the single adaptive filter facing a reference delay ramp of 0.01 sample/sample (note the scale difference with Fig. 6.56); $\mu = 0.01$

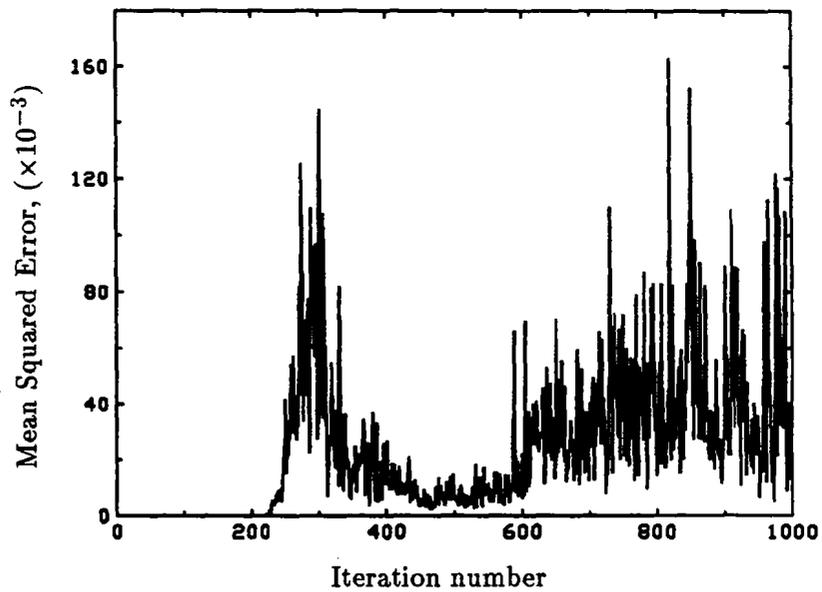


Fig. 6.58 Learning curve for the joint algorithm facing a sinusoidal reference delay (corresponding to Fig. 6.55); $\mu = 0.01$, $\alpha = 0.02$

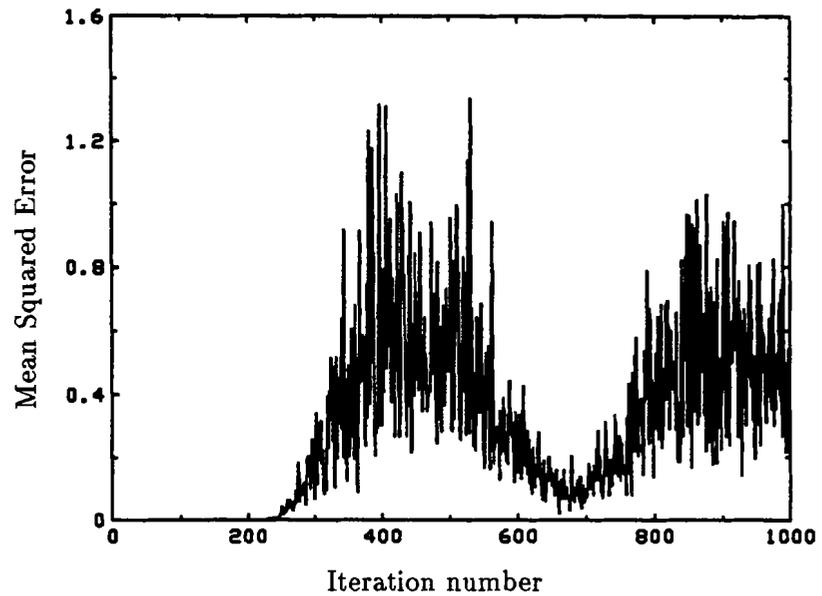


Fig. 6.59 Learning curve for the single adaptive filter facing a sinusoidal reference delay; $\mu = 0.01$ (note the factor of 10 compared to the scale of Fig. 6.58)

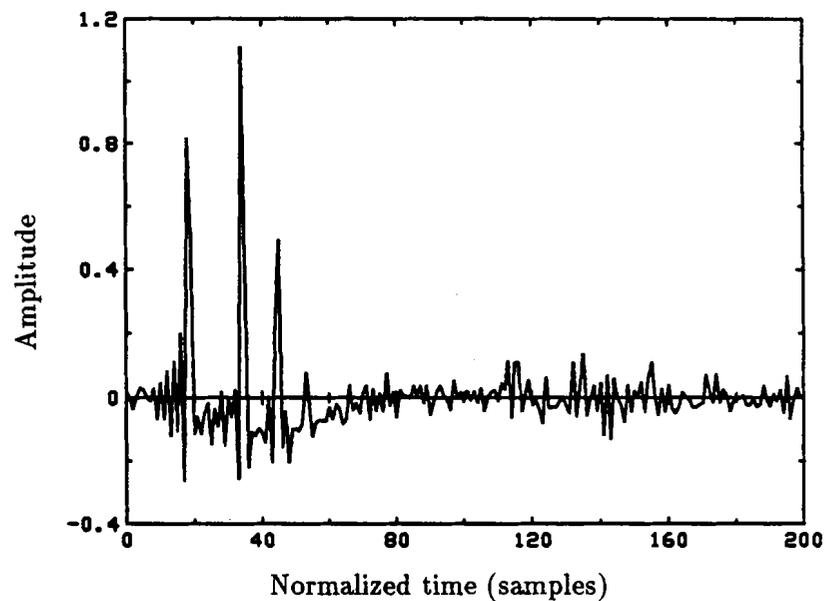


Fig. 6.60 Impulse response of the adaptive filter in the joint algorithm, after 1000 iterations, when the reference delay is a ramp of 0.01 sample/sample and $\mu = 0.01$, $\alpha = 0.02$

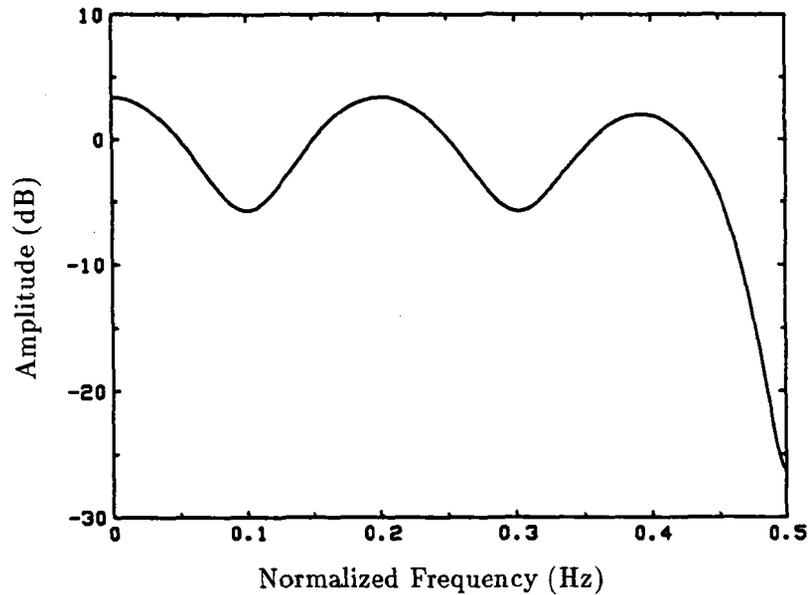


Fig. 6.61 Filter transfer function for coloured input generation

orders of magnitude, in order to compensate for the division by the fourth power. Therefore, $\alpha = 1000$ is used in the delay tracking simulations of a delay ramp and a sinusoidal delay. The results are illustrated in Figs. 6.65 and 6.66. Note that the tracking is good as long as the input amplitude is large, but that it becomes less accurate when the input samples size drops (around iteration 1400 on Figs. 6.65 and 6.66). Despite these problems, the normalized adaptive delay algorithm performs far better than the ordinary LMS algorithm of equation (4.33) when the input amplitude experiences large variations.

6.5.2 Results with a Joint Hybrid LMS Delay - RLS Filter in Type II-DRB

The joint RLS algorithm has been tested with the long reference filter impulse response used in Section 6.5.1 and illustrated in Fig. 6.53. Both the delay estimator and the adaptive weight vector give unsatisfactory results. By using the RLS adaptive filter alone, it was found that the filter could not track any of the linearly or sinusoidally changing reference delay that the shorter filter could easily follow before. This result was unexpected, since the tracking time constant of the RLS algorithm was derived to be [73], [33]

$$\tau_{LS} \approx \frac{1}{1 - \lambda},$$

which is independent of the number of adaptive filter coefficients. But in practice, it appears that the RLS adaptive filter is slowed down by an increase of its time span. Even a decrease

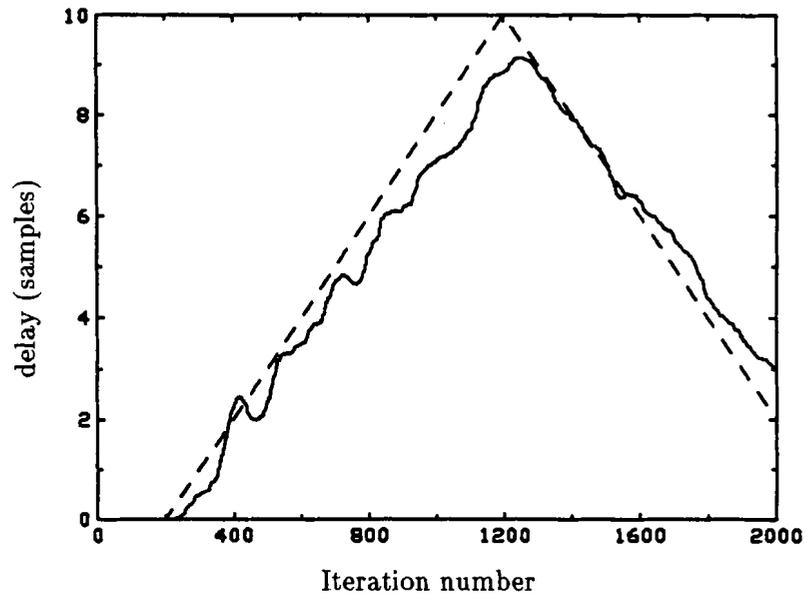


Fig. 6.62 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample, for a 200-tap reference impulse response and a coloured input; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$

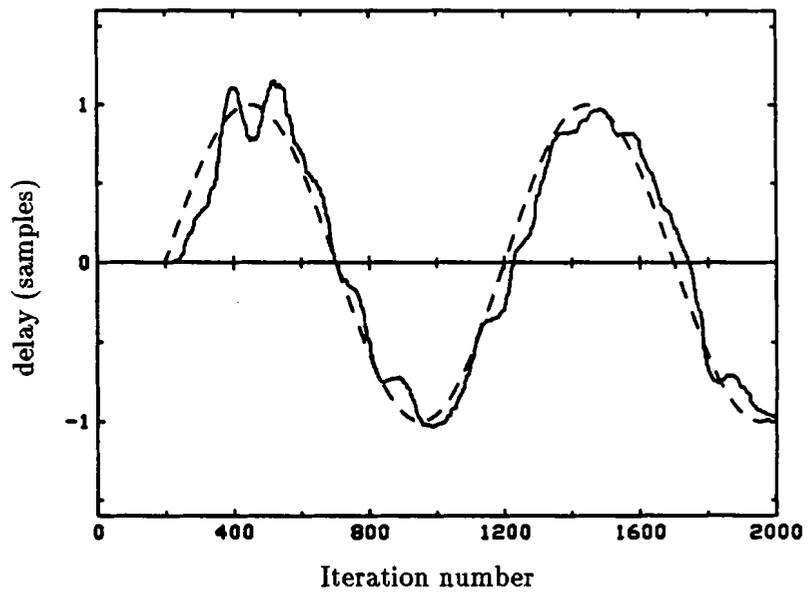


Fig. 6.63 LMS Adaptive delay response to a sinusoidal reference delay variation, for a 200-tap reference impulse response and a coloured input; dashed curve: reference delay; $\mu = 0.01$, $\alpha = 0.02$

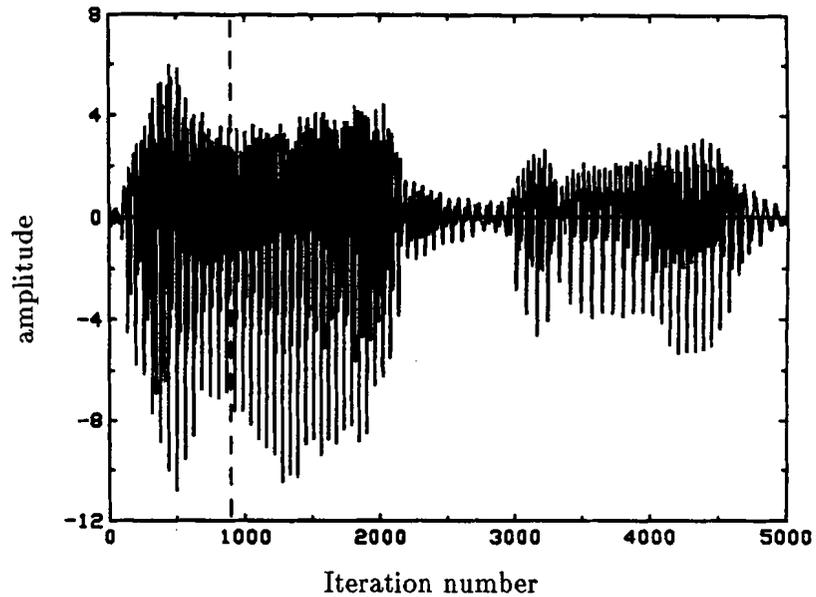


Fig. 6.64 Speech segment used for simulations; the dashed line indicates the range of data used for delay tracking

of the weighting factor β does not allow adequate tracking[†]. This is disastrous for the joint RLS algorithm derived in Chapter 4, since the delay estimation is based on the tracking, by the adaptive filter, of the delay reference variations.

In order to make the RLS adaptive filter solution viable, even in the presence of rapid reference delay variations, a hybrid adaptive system has been tested in Type II-DRB configuration. The delay estimation is performed by an adaptive delay element working in conjunction with an LMS adaptation algorithm of the form

$$d_{n+1} = d_n - 2\alpha\epsilon(n)\hat{r}(nT - d_n). \quad (6.26)$$

The adaptive filtering is performed with the fast RLS algorithm of Appendix F. The joint hybrid algorithm is therefore of the form of equation (4.180), with the obvious change in the weight vector adaptation.

The hybrid algorithm has been tested with a white Gaussian input and a speech input. During these tests, the numerical stability problem appeared again. It could not be solved as before, by the implementation of a parallel restart algorithm, because of the way the error signal is used in the LMS delay algorithm of (6.26). Recall that in the parallel restart algorithm, a parallel RLS algorithm is started from scratch on a regular basis, and its

[†] In fact, reducing the weighting factor increases the tendency for the RLS algorithm to become numerically unstable [68].

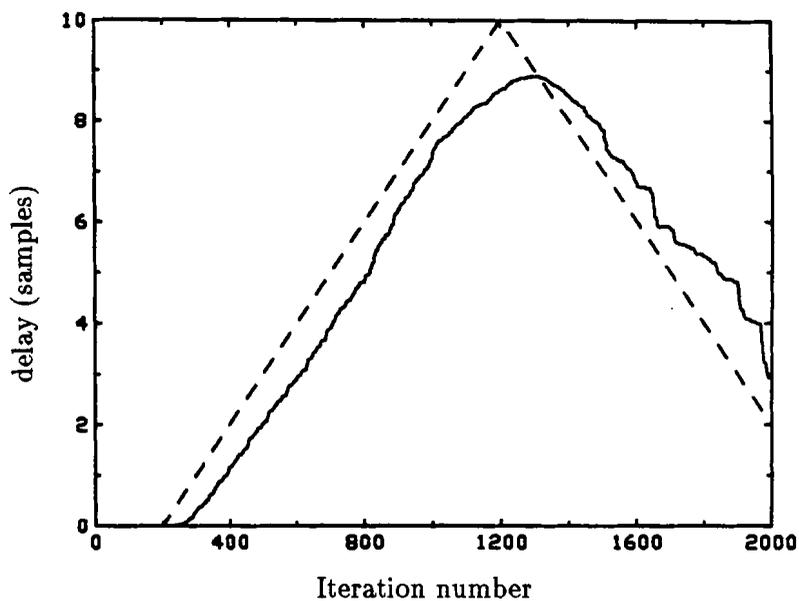


Fig. 6.65 Normalized LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample, for a 200-tap reference impulse response and a speech input; dashed curve: reference delay; $\mu = 10^{-5}$, $\alpha = 1000$

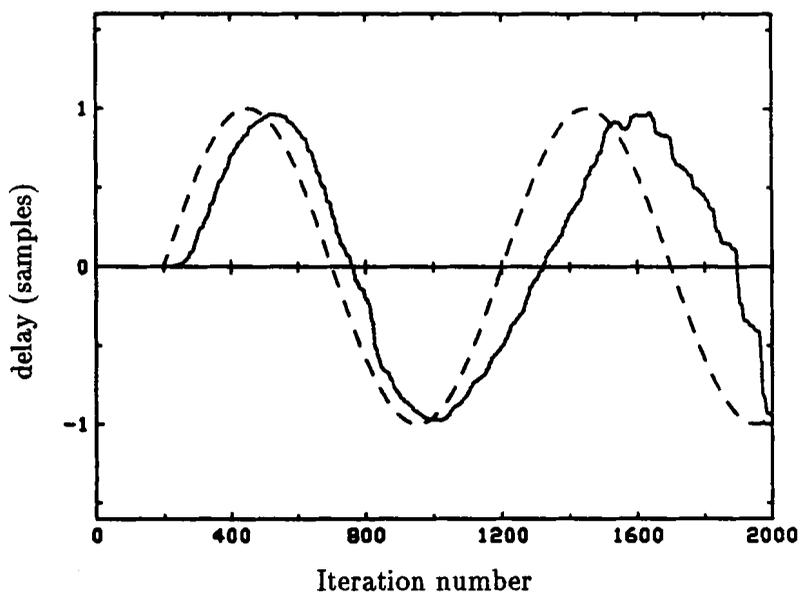


Fig. 6.66 Normalized LMS Adaptive delay response to a sinusoidal reference delay variation, for a 200-tap reference impulse response and a speech input; dashed curve: reference delay; $\mu = 10^{-5}$, $\alpha = 1000$

internal variables, as well as its weight vector, are transferred to the main RLS algorithm before numerical problems happen. This process, although very smooth, is not totally free of transition errors. The weight vector, before and after the transfer, is slightly different, which cause a certain jump in the error signal. This error burst is usually big enough to disturb greatly the LMS delay estimation and to cause the joint algorithm to lose track of the right estimates. Note that this problem did not appear in the joint RLS algorithm. No investigations were performed to find ways to overcome the instability problem, as it appears to be a fundamental limitation of the fast implementations of the RLS adaptive filter algorithm. The results given about the joint hybrid algorithm were therefore obtained before the instability appeared, and are good enough to illustrate the behaviour of the algorithm.

6.5.2.1 White Gaussian Input

The delay tracking by the joint hybrid algorithm is shown in Figs. 6.67 and 6.68, for a reference delay ramp and a sinusoidal reference delay in noiseless conditions.

Note the lag between the application of the reference delay and the response of the adaptive delay. This phenomenon was already noticed for the joint LMS algorithm in Type II-DRB. Note also that the difference between the reference delay ramp and the adaptive delay increases with time, and that the sinusoidal adaptive delay variations have an amplitude smaller than the reference delay variations. These discrepancies between the reference and the estimate delays are due to the adaptive filter action. Since the adaptive delay takes care of the biggest part of the reference delay, the variations seen by the adaptive filter are reduced accordingly, and they can be in part tracked by the RLS algorithm. The dramatic improvement of the joint hybrid algorithm over the single adaptive RLS filter, when rapid reference delay variations occur is illustrated by the learning curves of Figs. 6.69 and 6.70. Note the scale difference between these two figures.

6.5.2.2 Speech Input

The segment of speech used is again the one shown in Fig. 6.64. The RLS adaptive filter algorithm is essentially not unaffected by the eigenvalue spread of the input signal autocorrelation matrix [7], but the adaptive LMS delay has to be normalized as in Section 6.5.1. The results are illustrated in Figs. 6.71 and 6.72 for a reference delay ramp and a sinusoidal delay respectively in noiseless conditions. Note that, as in the case of the joint LMS algorithm with normalized delay, the delay tracking is good, but that the amplitude variations are nevertheless detrimental to the delay estimate quality.

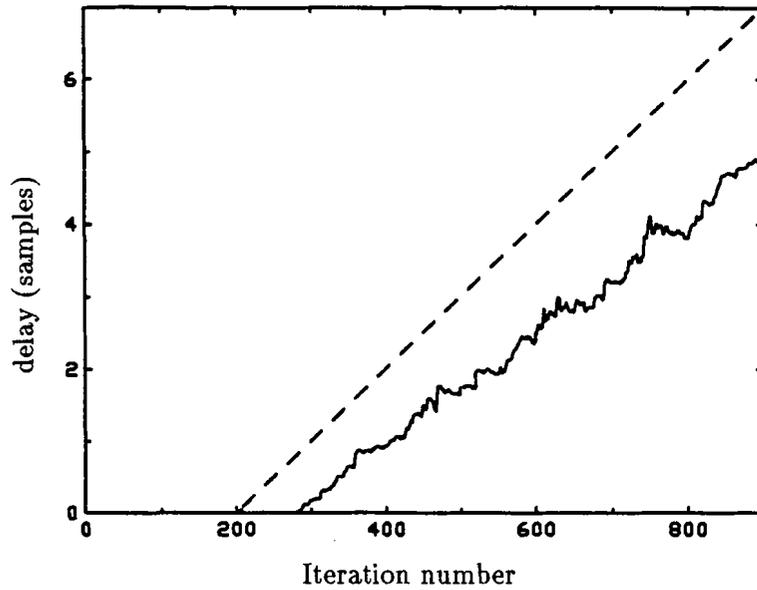


Fig. 6.67 LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 0.02$

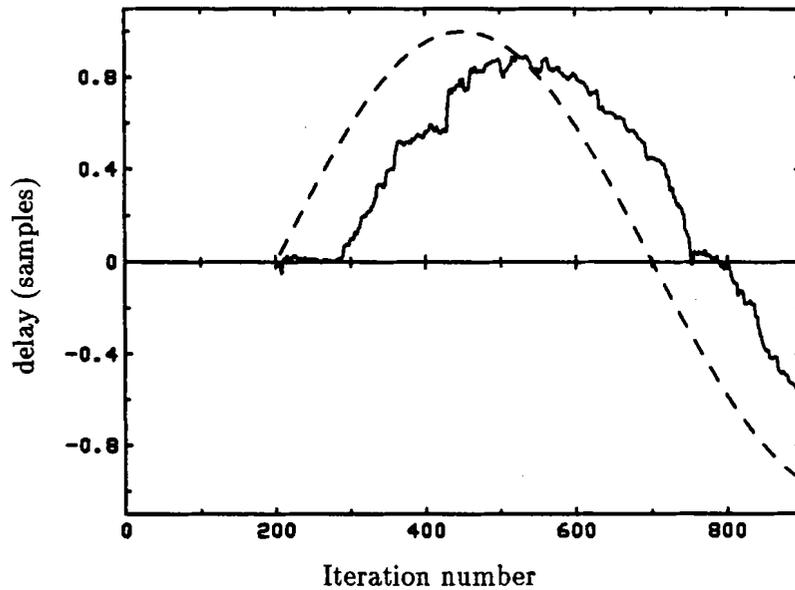


Fig. 6.68 LMS Adaptive delay response to a sinusoidal reference delay variation when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 0.02$

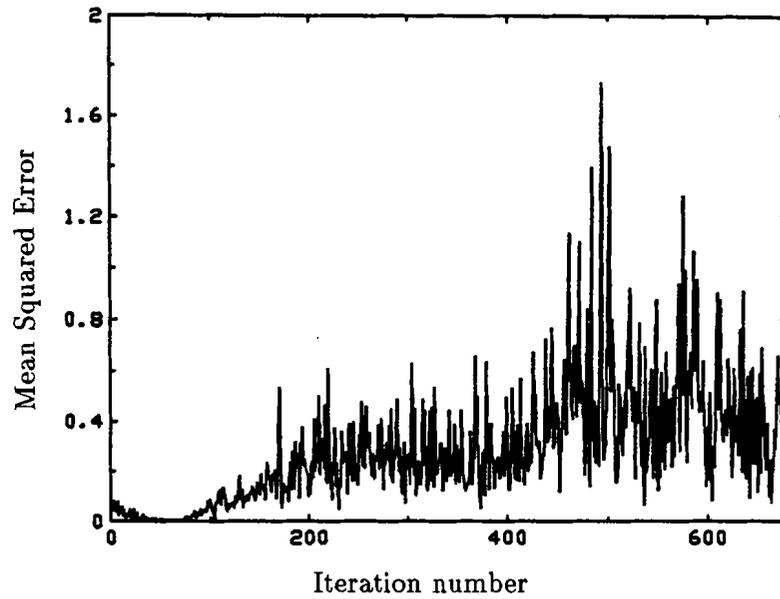


Fig. 6.69 Learning curve for the joint hybrid algorithm facing a delay ramp of 0.01 sample/sample; $\beta = 0.92$, $\alpha = 0.02$

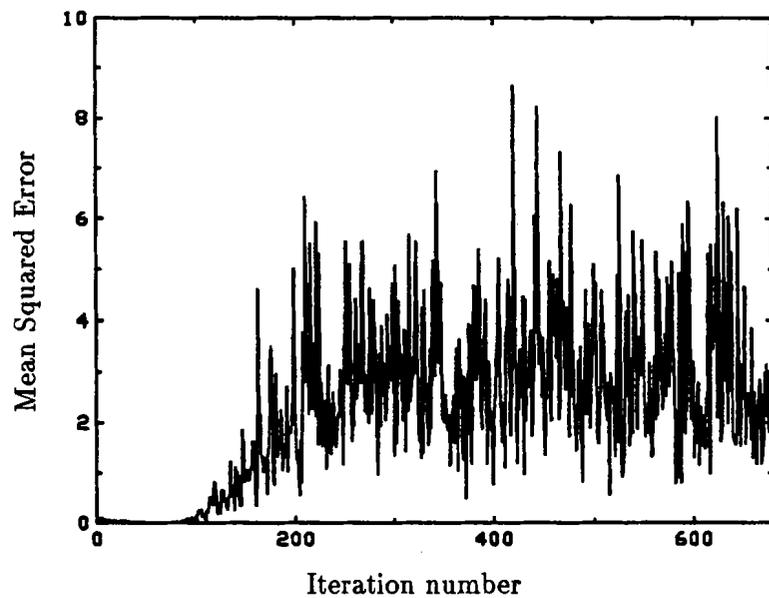


Fig. 6.70 Learning curve for the single adaptive filter facing a reference delay ramp of 0.01 sample/sample (note the scale difference with Fig. 6.69); $\beta = 0.92$

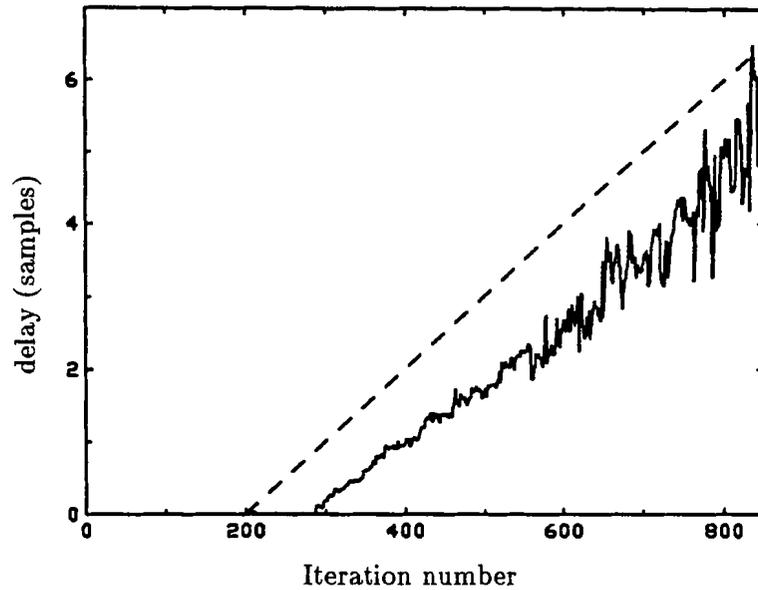


Fig. 6.71 Normalized LMS Adaptive delay response to a reference delay ramp of 0.01 sample/sample when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 2000$

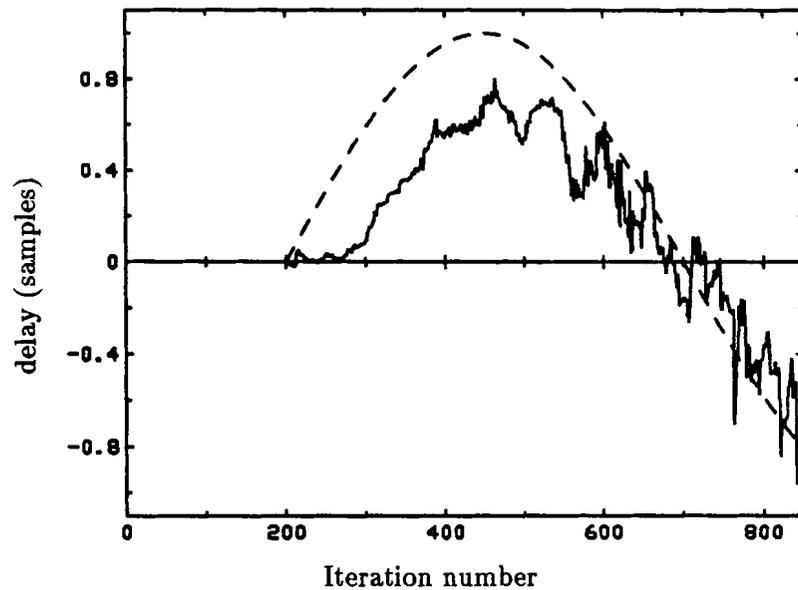


Fig. 6.72 Normalized LMS Adaptive delay response to a sinusoidal reference delay variation when the RLS adaptive filter has 200 coefficients; dashed curve: reference delay; $\beta = 0.92$, $\alpha = 2000$

6.6 Summary

Numerous experimental results about the joint LMS and the joint RLS algorithms were presented in this chapter. A typical reference filter was chosen, and white signals were utilized in most of the simulations. The joint LMS algorithm was considered first. The non-unique convergence property of the algorithm was illustrated and the theoretical delay tracking bounds were computed. Based on these results, the delay tracking capabilities of the algorithm were investigated, for a reference delay step and for a linearly and a sinusoidally changing reference delay. Both the Type I and the Types II configurations were considered, in noiseless and noisy conditions. The two types were compared together and it was found that the Type II-DAB tracks better the reference delay variations, while the Type II-DRB retards the adaptive delay response.

The theoretical results of Chapter 4 were computed and showed good agreement with the simulations. The tracking capabilities of the joint RLS algorithm were simulated for a short adaptive filter length. Both linearly and sinusoidally changing reference delays can be tracked, in noiseless and noisy conditions.

The joint LMS algorithm, with a typical reverberant room 200-tap impulse response, was simulated in Type I configuration, with white, coloured and speech inputs. A normalized LMS adaptive delay algorithm was used in the last case. The delay tracking characteristics are found to be adequate, even in these more practical examples. Finally, a joint hybrid algorithm, made of an LMS adaptive delay and an RLS adaptive filter, was considered when the number of coefficients in the filter is large. In this case, it was found that even the RLS algorithm cannot cope properly with rapid reference delay variations. The joint RLS algorithm is therefore not appropriate, and the addition of an LMS delay element allowed the use of the filter in these adverse conditions.

7.1 Summary

The work reported in this thesis represents a contribution to the subjects of adaptive time delay estimation and adaptive filtering. The conventional model used in time delay estimation is first enlarged, in order to include an unknown linear reference filter. The joint estimation problem is then formulated as a combined estimation of the delay and the reference filter. Two types of combined systems are to be estimated; the Type I system, in which the reference delay is located in front of the reference filter and the Type II configuration, where the delay follows the filter.

Three estimation criteria are first considered. The maximum likelihood (ML) estimator, for a finite observation interval and Gaussian signals, is derived in terms of a two-dimensional noncausal linear MMSE point estimator and of a bias term. This joint estimator is then specialized to the long observation interval case. The result is a new joint open-loop estimator involving time-invariant filters, which can be made causal and used as a suboptimal receiver for finite observation intervals. Closed-loop forms of this receiver are introduced and discussed. It is concluded that the form obtained for the ML estimator is not well suited for a practical application. But this form is instructive in that it is composed of a delay element, in series with a group of filters derived from the estimate of the reference filter. The structure of the joint MMSE and LS estimators is then introduced. It retains the delay-filter form of the ML estimator, and is composed of an adaptive delay element in series with an adaptive filter. The estimation criterion is used to minimize a function of the squared error between the joint adaptive system and the reference system outputs.

The first derivative-based joint algorithm considered is the Steepest-Descent (SD) algorithm. In this algorithm, the adaptive delay element is adjusted in the direction opposite to

the derivative of the MSE function with respect to the delay. The filter adaptation algorithm is the conventional SD algorithm, in which the filter response is adapted in the direction opposite to the gradient of the MSE function with respect to weight vector. The MSE function is derived and is shown to be related to both the adaptive filter and the reference filter impulse responses, as well as to the input signal power spectral density. This typically causes the performance surface to be multimodal with respect to the adaptive delay value. A closed-loop derivative-based delay estimation is therefore subject to convergence to local solutions. In the weight vector subspace, the convergence is unimodal since the MSE function is quadratic with respect to the weight vector. It is shown that when the gradient with respect to the weight vector is zero, this corresponds to a necessary and sufficient condition for convergence of the joint SD algorithm. This implies that the joint algorithm suffers also from non-unique solutions in the joint weight vector-delay vector space.

The joint SD algorithm being composed of two adaptation algorithms, the alternation of the two processes changes the convergence characteristics. For a joint algorithm which alternates its two components in any fashion, simple conditions for convergence on the two gain factors μ and α are found. The bound on the filter gain factor μ is identical to the one for the usual SD adaptive filter. It is equal to the inverse of the maximum eigenvalue of the input signal autocorrelation matrix. The bound on the delay gain factor is shown to be such that α must be smaller than twice the inverse of the MSE function second derivative, evaluated at the closest minimum. If the delay value is close to the optimum solution, than α must be smaller than twice the inverse of $\ddot{\xi}_{\min}$. It is also derived that, in tracking conditions, this second derivative is also inversely proportional to the delay time constant of adaptation. It is demonstrated that the gain factors can be related to each other by applying a constraint on the relative speed of convergence of the two adaptive processes. The constraint is such that the adaptive delay is faster than the adaptive filter.

The joint Least-Mean-Square (LMS) algorithm is then presented as a stochastic implementation of the joint SD algorithm. This algorithm is defined by replacing the MSE function by the squared error in the SD algorithm. Three versions of the joint LMS algorithm are shown to be of interest. The Type I configuration mimics the reference system of the same type. The Type II-DAB form reproduces the Type II reference system where the delay is located directly after the filter. The Type II-DRB estimates a Type II reference system by using a negative adaptive delay in the reference branch. It is shown, by using the ODE method, that if the adaptation factors are time-variant and both tend toward zero, the joint LMS algorithm converges to a local minimum of the MSE function, like the exact version of the joint SD algorithm. This result confirms the conjecture that if the adaptation

factors are small enough, the joint LMS algorithm and the joint SD algorithms tend to similar solutions.

Using a series of commonly made assumptions, the conditions on each gain factors, for convergence in the mean and in the mean square, are derived for the three types of configurations. It is found that the bounds on α and μ , for convergence in the mean of the LMS estimates, are identical to the bounds for the SD estimates in every type of configuration. The bounds on α , for delay convergence in the mean square, are functions of the ratio between $\ddot{\xi}_{\min}$ and $E[G_n^2]$, a quantity that is a function of the input signal power, the second derivative $\ddot{\xi}_{\min}$, the reference power and the variance in the adaptive filter weight vector estimate. The LMS delay estimate is shown to be unbiased and its variance is derived to be a function of $\ddot{\xi}_{\min}$ and $E[G_n^2]$, as well as a function of the variance of the delay derivative noise estimate. The weight vector estimate is shown to be biased by a vector proportional to the delay estimate variance and inversely proportional to the input signal autocorrelation matrix. In Type I and Type II-DRB configurations, the condition for convergence in the mean square of the weight vector estimate is found to be identical to the usual condition for a single adaptive filter, i.e. μ must be lower than the inverse of the trace of the input signal autocorrelation matrix. For the Type II-DAB, the condition is more complicated, but it is also identical with and without the adaptive delay. In all the configurations, the trace of the weight noise vector correlation matrix is found to be proportional to the MMSE, to the second derivative of the MSE function at its minimum and to the delay estimate variance. The expressions for the excess MSE and for the misadjustment associated to the joint LMS algorithm are derived. In every type of configuration, these expressions are shown to be equal to the sum of three terms; a term specific to the delay estimate, a term specific to the adaptive filter and a cross-product term related to both estimates. The cross-product misadjustment is equal to the product of the two specific misadjustments. Among the three types of joint configurations, the Type II-DAB is found to be the less appealing. The location of the delay, after the adaptive filter, limits the tracking ability of the filter by reducing the stability bound on μ , and increases the excess MSE.

For faster tracking of reference variations, the joint recursive least squares algorithm is presented. It is based on the least squares (LS) estimation criterion and minimizes the sum of exponentially weighted squared errors, with respect to both the integer delay estimate, defined as the “lag”, and the weight vector. Because of the short convergence time of the RLS filter algorithm, the delay estimation and the adaptive filtering parts of the joint algorithm have to be intimately linked to each other. This task is done by first computing the RLS adaptive filter, and then by “extracting” the delay information from

the resulting error and weight vector. Two joint RLS algorithms are derived and exploit the data structure, in order to compute the adaptive weight vector and the lag value, within a finite set, corresponding to the joint LS solution. In order to perform such a task, the sum of squared errors is computed for each value of the integer delay estimate in the set of interest, and the delay value corresponding to the lowest value is retained. This is accomplished by using a series of lag-recursive relations that allows the efficient computation, based on the LS solution for the current lag, of the sum of squared errors for other values of the lag. These lag-recursive relations are derived, for both a Type I and a Type II-DRB configurations, by using a geometrical approach, and are appended to the fast transversal filter (FTF) adaptive filtering algorithm, in order to form the joint RLS algorithm. This new algorithm is composed of three distinct phases. The first one involves the update of the forward and backward linear predictors used in both the FTF and in the lag-recursive relations. The second phase involves the use of the lag-recursive relations, in order to compute the current optimum weight vector and to derive the sums of squared errors for the lags comprised in the set of interest. The third computational phase involves a decision on the lag update and the computations, in the case of update, of the new corresponding variables. This last task is made easier by the use of some of the lag-recursive relations. This new joint algorithm exploits fully the lag recursions in order to allow the serial computation, from a single set of stored weight vector and error variables, of the information necessary for the decisions about the lag update.

The analysis of the joint RLS algorithm shows that the delay adaptation process is characterized by a discrete-time Markov chain, which renders the analysis difficult. Under the assumptions used in the analysis of the joint LMS algorithm, the LS delay estimator is shown to be unbiased, while the weight vector estimator is biased by the same quantity found in the joint LMS algorithm. The weight noise vector correlation matrix is found to be proportional, as in its joint LMS counterpart, to the MMSE, to the second derivative of the MSE function at its minimum and to the delay estimate variance. The expression for the excess MSE is derived to be also composed of three terms, bearing a form very similar to the form found in the LMS algorithm.

The joint LMS and RLS algorithms are then simulated. The experimental set-up is that of a system identification (cancellation) configuration. Spectrally white signals, as well as coloured and speech inputs are considered. A short reference impulse response is used, as well as a longer one, typical of a reverberant room. The delay estimation of time-varying reference delays is illustrated, for both linearly and sinusoidally changing conditions and for noiseless and noisy cases. For the short reference impulse response, the LMS adaptive filter

can compensate for some of the reference delay variations, while for the long response, the adaptive delay allows a considerable reduction of the mean squared error. For the case of a speech input, a normalized form of the LMS adaptive delay is introduced, in order to cope with the large input amplitude variations. In the joint RLS algorithm, the delay estimation allows the adaptive filter to stay “centered” and to better model the reference filter. For a long adaptive filter impulse response, a hybrid LMS delay-RLS filter is defined and reduces considerably the mean squared error.

The analyses and the simulations of the joint LMS and RLS algorithms demonstrate the ability of the joint techniques to improve upon the performances of the conventional methods, when there is a relative delay between the main input and the reference signal. In general, the joint algorithms produce a lower mean squared error between their outputs and the reference signal. Furthermore, they allow the use of adaptive filters with a smaller number of coefficients.

7.2 Contributions

This thesis has contributed to the theories of delay estimation and of adaptive digital filtering, as well as to the field of joint adaptive algorithms. The major contributions of this work can be summarized as follows:

1. The joint maximum likelihood estimator for a reference delay and a reference filter has been derived for Gaussian signals, using both a finite and an infinite observation interval. This estimator has been used to define the structure of the joint MSE and LS adaptive estimators.
2. The joint steepest-descent and least-mean-square adaptive algorithms, composed of an adaptive delay element and of an adaptive transversal filter, have been analysed [41]. These algorithms constitute the generalizations of existing gradient-based time delay estimation algorithms without reference filter. They can also be regarded as upgrades of the conventional SD or LMS adaptive filter algorithms, since they allow the synchronization, in a general framework, of the input and the reference signals used by an adaptive filter. The joint LMS algorithm has been implemented and tested under various conditions.
3. The interaction between the LMS adaptive delay and the LMS adaptive filter estimates has been derived for three types of delay and filter arrangements. The joint excess MSE expression was shown to be a function of three terms; one term specific to the adaptive delay, one term specific to the adaptive filter and one cross-product term related to both estimates. Experiments have confirmed the form of the MSE expression.
4. An existing set of block-based lag-recursive relations has been extended to a set of on-line relations. A new geometrical derivation has been used to obtain and

interpret these relations. This set of relations allows the serial computation, from an initial value of the RLS solution at a certain lag, of the LS weight vector and the corresponding sum of squared errors for other lag values. These relations have been verified by simulations.

5. A new type of joint adaptive delay and adaptive filter RLS algorithm has been designed by appending the lag-recursive relations to the fast transversal RLS filter algorithm and by using a serial computation of the critical parameters used for lag update [42]. This algorithm has been implemented and tested for different conditions.
6. The joint RLS algorithm has been shown to produce an excess MSE bearing a great resemblance with the excess MSE produced by the joint LMS algorithm.
7. For applications where large adaptive filters are required, the joint algorithms have been shown to produce a significantly lower mean squared error. A hybrid joint algorithm, formed of an RLS adaptive filter and an LMS adaptive delay, has been successfully implemented for that purpose.

7.3 Future Work

The following points could constitute the basis for future research.

1. It has been assumed, throughout this thesis, that the delay estimate is close enough to the global minimum of the MSE function such that convergence to this minimum happens. This assumption, although common in the delay estimation literature, is not necessarily true in practice. Some form of delay acquisition procedure is necessary and should be studied. The LS estimation criterion could be used for that purpose by observing a block of input signal, and by applying an algorithm similar to the joint RLS algorithm, for an extensive set of possible lags. This *optimum lag* algorithm has been proposed in [63] and could constitute a parallel processor of the form proposed in [43, p. 279], for minimum searching of a multimodal function.
2. The problem of multitude convergence points and false lock of the delay estimator has to be studied and solved. One solution is to periodically realign the adaptive filter input and reference by acquiring a delay estimate close to the optimum. This could be done off-line, by using a procedure similar to the one proposed for acquisition.
3. In the joint SD or LMS algorithms, a higher order delay loop could be used to speed up the convergence rate.
4. Data reuse could be implemented in the joint SD or LMS algorithms by repeating one of the two adaptive processes on the same input vector, as proposed in Section 3.3.
5. The joint SD and LMS algorithms could merge in some manner the two adaptive processes. For example, the interpolator implementing the fractional delay element could be incorporated into the adaptive filter. This would create a new class of joint adaptive algorithm.
6. The possibility of implementing a computationally efficient joint RLS algorithm with a fractional delay estimator could be investigated.

7. The numerical stability of the RLS adaptive filter algorithm has to be reconsidered, in light of its influence on the delay estimation.

Appendix A. Derivation of the Joint Maximum Likelihood Estimator For a Type I System

Based on a vector mathematical model, the form of the joint ML estimator, over an interval $[n_1, n_2]$, is derived. The likelihood probability and the likelihood function are computed in Section A.1. The likelihood function is shown to be the sum of a noncausal term $\ell_Y(d, \mathbf{w})$ and a bias term $\ell_B(d, \mathbf{w})$. As noticed in Section 2.3.1, the function $\ell_Y(d, \mathbf{w})$ is expressed in terms of MMSE estimation. The MMSE estimator necessary to compute $\ell_Y(d, \mathbf{w})$ is explicitly derived in Section A.2. The function $\ell_Y(d, \mathbf{w})$ is computed for long observation intervals in Section A.3. The material presented in this appendix is an extension of the work reported by Stuller in [16]. The extension is done for a reference branch including a linear filter, and the results are given here for discrete-time signals and systems. Most of the derivations follow closely Stuller's procedures, and it would make the reading easier if his article would be consulted from time to time.

A.1 Derivation of The Log-Likelihood Function

The derivation is based on the mathematical model of equations (2.7) to (2.10). These equations are reproduced here for convenience.

$$\mathbf{y}(n) = \mathbf{s}(n|d, \mathbf{w}) + \mathbf{v}(n), \quad (\text{A.1})$$

$$\mathbf{y}(n) = \begin{bmatrix} y_1(n) \\ y_2(n) \end{bmatrix} \quad (\text{A.2})$$

$$\mathbf{s}(n|d, \mathbf{w}) = \begin{bmatrix} s(n) \\ \mathcal{L}_{d, \mathbf{w}(n)}^I[s(n)] \end{bmatrix} \quad (\text{A.3})$$

$$\mathbf{v}(n) = \begin{bmatrix} v_1(n) \\ v_2(n) \end{bmatrix}. \quad (\text{A.4})$$

Based on these vector definitions, the log-likelihood function is derived as in Stuller. First of all, the received vectors $\mathbf{y}(n)$ is expressed as an infinite-dimensional vector \mathbf{y} , using the discrete-time normalized vector eigenfunctions $\mathbf{f}_i(n|d, \mathbf{w})$ of the input signal covariance matrix $\Phi_{ss}(k|d, \mathbf{w})$, over an observation interval $[n_1, n_2]$, i.e.

$$\mathbf{y}(n) = \lim_{N \rightarrow \infty} \sum_{i=1}^N y_i \mathbf{f}_i(n|d, \mathbf{w}), \quad (\text{A.5})$$

where

$$y_i = \sum_{n=n_1}^{n_2} \mathbf{f}_i^H(n|d, \mathbf{w}) \mathbf{y}(n). \quad (\text{A.6})$$

The covariance matrix is defined as

$$\Phi_{ss}(k|d, \mathbf{w}) = E[\mathbf{s}(n+k|d, \mathbf{w})\mathbf{s}^H(n|d, \mathbf{w})], \quad (\text{A.7})$$

and the normalized vector eigenfunctions are 2×1 column vectors satisfying the equations

$$\lambda_i(d, \mathbf{w})\mathbf{f}_i(n|d, \mathbf{w}) = \sum_{m=n_1}^{n_2} \Phi_{ss}(n-m|d, \mathbf{w})\mathbf{f}_i(m|d, \mathbf{w}) \quad (\text{A.8})$$

$$\sum_{n=n_1}^{n_2} \mathbf{f}_i^H(n|d, \mathbf{w})\mathbf{f}_j(n|d, \mathbf{w}) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases} \quad (\text{A.9})$$

Note that $\lambda_i(d, \mathbf{w})$ is the *scalar* eigenvalue associated with $\mathbf{f}_i(n|d, \mathbf{w})$. It is assumed that the covariance matrix $\Phi_{ss}(k|d, \mathbf{w})$ is a positive definite function, i.e. that [43]

$$\sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{f}^H(n)\Phi_{ss}(n-m|d, \mathbf{w})\mathbf{f}(m) > 0, \quad (\text{A.10})$$

for any vector $\mathbf{f}(n)$ with finite energy over $[n_1, n_2]$ [†]. In this case, all the eigenvalues are real and strictly positive numbers, and the set of eigenfunctions is a complete orthonormal set over the interval $[n_1, n_2]$, i.e.

$$\lim_{N \rightarrow \infty} \sum_{n=n_1}^{n_2} \left[\mathbf{x}(n) - \sum_{i=1}^N x_i \mathbf{f}_i(n|d, \mathbf{w}) \right]^2 = 0, \quad (\text{A.11})$$

for any finite energy deterministic vector function $\mathbf{x}(n)$ over $[n_1, n_2]$ and

$$\lim_{N \rightarrow \infty} E \left[\left(\mathbf{u}(n) - \sum_{i=1}^N u_i \mathbf{f}_i(n|d, \mathbf{w}) \right)^2 \right] = 0, \quad (\text{A.12})$$

for any finite energy random function $\mathbf{u}(n)$ over $[n_1, n_2]$.

Therefore, all the information present in $\mathbf{y}(n)$ is present in the vector $\mathbf{y}_N = [y_1, y_2, \dots, y_N]$ for N tending to infinity. Given the parameters d and \mathbf{w} , the y_i 's are independent zero-mean Gaussian random variables with variance

$$E[|y_i|^2|d, \mathbf{w}] = \lambda_i(d, \mathbf{w}) + N_o/2. \quad (\text{A.13})$$

[†] A finite energy vector function $\mathbf{x}(n)$ over $[n_1, n_2]$ is such that

$$\sum_{n=n_1}^{n_2} \mathbf{x}^H(n)\mathbf{x}(n) < \infty.$$

The joint probability density function of the y_i 's, for $1 \leq i \leq N$, and given the parameters d and \mathbf{w} , is therefore

$$p_{\mathbf{Y}_N|D, \mathbf{W}}(\mathbf{y}_N|d, \mathbf{w}) = \left[\prod_{i=1}^N \frac{1}{\sqrt{2\pi[\lambda_i(d, \mathbf{w}) + N_o/2]}} \right] \exp \left[-\frac{1}{2} \sum_{i=1}^N \frac{|y_i|^2}{\lambda_i(d, \mathbf{w}) + N_o/2} \right]. \quad (\text{A.14})$$

The likelihood function $\ell'_N(d, \mathbf{w})$ is obtained by taking the logarithm of $p_{\mathbf{Y}_N|D, \mathbf{W}}(\mathbf{y}_N|d, \mathbf{w})$, when N tends toward infinity, and by making use of equation (A.6), which gives

$$\begin{aligned} \ell'(d, \mathbf{w}) = & -\frac{1}{2} \sum_{i=1}^{\infty} \ln[2\pi\{\lambda_i(d, \mathbf{w}) + N_o/2\}] \\ & -\frac{1}{2} \sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{y}^H(n) \sum_{i=1}^{\infty} \frac{\mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w})}{\lambda_i(d, \mathbf{w}) + N_o/2} \mathbf{y}(m). \end{aligned} \quad (\text{A.15})$$

Define the inner sum as $Q(n, m|d, \mathbf{w})$, for $n_1 \leq n \leq n_2, n_1 \leq m \leq n_2$. This function can be expanded as

$$\begin{aligned} Q(n, m|d, \mathbf{w}) &= \sum_{i=1}^{\infty} \frac{\mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w})}{\lambda_i(d, \mathbf{w}) + N_o/2} \\ &= \frac{2}{N_o} \sum_{i=1}^{\infty} \left[\mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w}) - \frac{\lambda_i(d, \mathbf{w})}{\lambda_i(d, \mathbf{w}) + N_o/2} \mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w}) \right] \\ &= \frac{2}{N_o} \delta(n-m) \mathbf{I} - \frac{2}{N_o} \sum_{i=1}^{\infty} \frac{\lambda_i(d, \mathbf{w})}{\lambda_i(d, \mathbf{w}) + N_o/2} \mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w}), \end{aligned} \quad (\text{A.16})$$

for $n_1 \leq n \leq n_2, n_1 \leq m \leq n_2$. Defining the function $Q_2(n, m|d, \mathbf{w})$ as

$$Q_2(n, m|d, \mathbf{w}) = \sum_{i=1}^{\infty} \frac{\lambda_i(d, \mathbf{w})}{\lambda_i(d, \mathbf{w}) + N_o/2} \mathbf{f}_i(n|d, \mathbf{w}) \mathbf{f}_i^H(m|d, \mathbf{w}), \quad (\text{A.17})$$

for $n_1 \leq n \leq n_2, n_1 \leq m \leq n_2$, the likelihood function of (A.15) can be written as

$$\begin{aligned} \ell'(d, \mathbf{w}) &= -\frac{1}{2} \sum_{i=1}^{\infty} \ln[2\pi\{\lambda_i(d, \mathbf{w}) + N_o/2\}] \\ &+ \frac{1}{N_o} \sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{y}^H(n) Q_2(n, m|d, \mathbf{w}) \mathbf{y}(m) \\ &- \frac{1}{N_o} \sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{y}^H(n) \mathbf{y}(m) \delta(n-m). \end{aligned} \quad (\text{A.18})$$

The likelihood of (A.18) can be simplified by dropping the last term and adding the term $\ln[\sqrt{\pi N_o}]$ since none of these terms depends on the estimates. This finally gives the desired likelihood function

$$\ell(d, \mathbf{w}) = \ell_Y(d, \mathbf{w}) + \ell_B(d, \mathbf{w}), \quad (\text{A.19})$$

where

$$\ell_Y(d, \mathbf{w}) = \frac{1}{N_o} \sum_{n=n_1}^{n_2} \sum_{m=n_1}^{n_2} \mathbf{y}^H(n) Q_2(n, m|d, \mathbf{w}) \mathbf{y}(m) \quad (\text{A.20})$$

and

$$\ell_B(d, \mathbf{w}) = -\frac{1}{2} \sum_{i=1}^{\infty} \ln \left[\frac{2\lambda_i(d, \mathbf{w})}{N_o} + 1 \right]. \quad (\text{A.21})$$

In (A.20), $Q_2(n, m|d, \mathbf{w})$ is the matrix impulse response of the noncausal linear MMSE point estimator of $\mathbf{s}(n|d, \mathbf{w})$, from the received vector $\mathbf{y}(n)$, given the parameters d and \mathbf{w} [43]. It is given by the solution of the “normal” equation

$$\frac{N_o}{2} Q_2(n, m|d, \mathbf{w}) + \sum_{k=n_1}^{n_2} Q_2(n, k|d, \mathbf{w}) \Phi_{ss}(k - m|d, \mathbf{w}) = \Phi_{ss}(n - m|d, \mathbf{w}), \quad (\text{A.22})$$

for $n_1 \leq n \leq n_2, n_1 \leq m \leq n_2$. The form of the estimator is given in Fig. A.1.

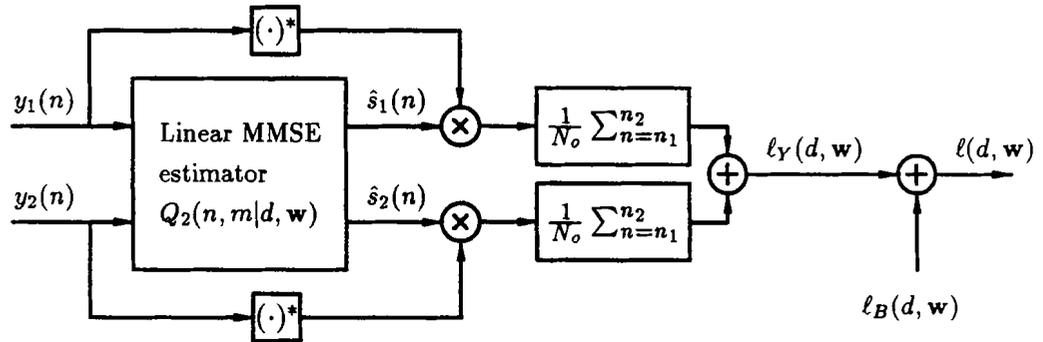


Fig. A.1 Blockdiagram of the noncausal joint maximum likelihood estimator (canonical realization number 1)

A.2 Derivation of Entries of $Q_2(n, m|d, \mathbf{w})$

The form of the entries of $Q_2(n, m|d, \mathbf{w})$ are derived, for an observation interval $[n_1, n_2]$, by using Stuller's constructive method [16]. The first step in the derivation of $Q_2(n, m|d, \mathbf{w})$ is to noncausally transform the received vector $\mathbf{y}(n)$ into a new vector $\mathbf{r}(n)$. The transformation is linear and invertible and, by the reversibility theorem, does not affect the performance of the system [43]. Its role is to transform the received vector, assuming the parameters d and \mathbf{w} , into a 2×1 vector $\mathbf{r}(n)$ in which the second component does not depend on the

transmitted signal $s(n)$ and the first component does. A transformation that accomplishes this task is

$$\mathbf{r}(n) = \begin{cases} \begin{bmatrix} \mathcal{L}_{d,\mathbf{w}}^{-1}[y_2(n)] \\ 0 \end{bmatrix}, & n_1 - \lfloor d/T \rfloor \leq n < n_1 \\ \frac{1}{2} \begin{bmatrix} y_1(n) + \mathcal{L}_{d,\mathbf{w}}^{-1}[y_2(n)] \\ y_1(n) - \mathcal{L}_{d,\mathbf{w}}^{-1}[y_2(n)] \end{bmatrix}, & n_1 \leq n < n_2 - \lfloor d/T \rfloor \\ \begin{bmatrix} y_1(n) \\ 0 \end{bmatrix}, & n_2 - \lfloor d/T \rfloor \leq n \leq n_2. \end{cases} \quad (\text{A.23})$$

Therefore, the vector $\mathbf{r}(n)$ takes the explicit form

$$\mathbf{r}(n) = \begin{cases} \mathbf{0}, & n < n_1 - \lfloor d/T \rfloor \\ \begin{bmatrix} s(n) \\ 0 \end{bmatrix} + \begin{bmatrix} z_1(n) \\ z_2(n) \end{bmatrix}, & n_1 - \lfloor d/T \rfloor \leq n \leq n_2 \\ \mathbf{0}, & n_2 < n, \end{cases} \quad (\text{A.24})$$

where

$$\begin{bmatrix} z_1(n) \\ z_2(n) \end{bmatrix} = \begin{cases} \begin{bmatrix} \mathcal{L}_{d,\mathbf{w}}^{-1}[v_2(n)] \\ 0 \end{bmatrix}, & n_1 - \lfloor d/T \rfloor \leq n < n_1 \\ \frac{1}{2} \begin{bmatrix} v_1(n) + \mathcal{L}_{d,\mathbf{w}}^{-1}[v_2(n)] \\ v_1(n) - \mathcal{L}_{d,\mathbf{w}}^{-1}[v_2(n)] \end{bmatrix}, & n_1 \leq n < n_2 - \lfloor d/T \rfloor \\ \begin{bmatrix} v_1(n) \\ 0 \end{bmatrix}, & n_2 - \lfloor d/T \rfloor \leq n \leq n_2. \end{cases} \quad (\text{A.25})$$

Note that $\mathcal{L}_{d,\mathbf{w}}^{-1}[\cdot]$ is defined as (for a Type I reference system)

$$\begin{aligned} \mathcal{L}_{d,\mathbf{w}}^{-1}[y(n)] &= w^{-1}(n) \otimes y(nT + d) \\ &= \sum_i w^{-1}(i) y(nT - iT + d) \\ &= \sum_i w^{-1}(nT - iT + d) y(i), \end{aligned} \quad (\text{A.26})$$

where $w^{-1}(n)$ is the impulse response of the inverse filter corresponding to $w(n)$, i.e.

$$w(n) \otimes w^{-1}(n) = \delta(n). \quad (\text{A.27})$$

At this point, the noncausal linear MMSE point estimate $\hat{\mathbf{s}}(n|d, \mathbf{w})$ from $\mathbf{r}(n)$ is wanted. A variant of Stuller's theorem [16] is invoked to perform this task.

Theorem. Assume a signal model of the form of equation (A.1), with

$$E[\mathbf{v}(n)\mathbf{v}^H(m)] = \frac{N_o}{2}\mathbf{I}\delta(n-m).$$

Assume that the linear invertible transformation of equation (A.23) is applied on $\mathbf{y}(n)$ and gives $\mathbf{r}(n)$. Then, the discrete-time noncausal linear MMSE point estimator of $s(n)$ from $\mathbf{r}(n)$, $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$, conditioned on the parameters d and \mathbf{w} , is given by the system of Figure A.2, where $f(n, m|d, \mathbf{w})$ is the impulse response of the noncausal linear MMSE point estimator of $z_1(n)$ from $z_2(n)$ and $g(n, m|d, \mathbf{w})$ is the impulse response of the noncausal linear MMSE point estimator of $s(n)$ from $s(n) + z_1(n) - \hat{z}_1(n)$. ■

The proof of this theorem is identical, *mutatis mutandis*, to the proof given in [16].

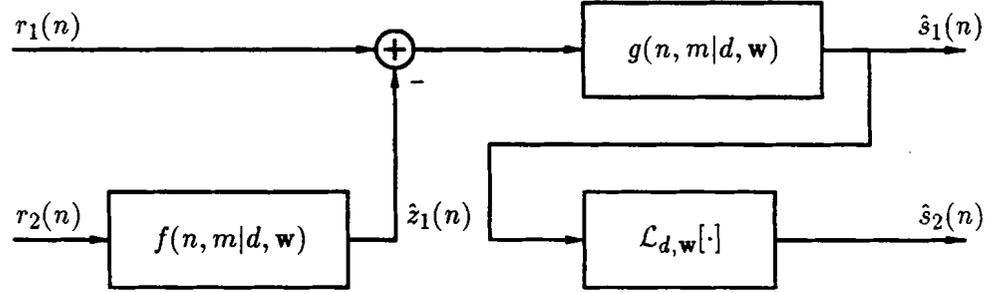


Fig. A.2 Structure of the discrete-time noncausal linear MMSE point estimator of $s(n)$ from $\mathbf{r}(n)$, $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$, conditioned on the parameters d and \mathbf{w} , as defined in the Theorem.

The outputs of these two linear MMSE estimators are given by

$$\hat{z}_1(n) = \sum_{m=n_1-\lfloor d/T \rfloor}^{n_2} f(n, m|d, \mathbf{w})z_2(m), \quad n_1 - \lfloor d/T \rfloor \leq n \leq n_2 \quad (\text{A.28})$$

and

$$\hat{s}_1(n) = \hat{s}(n) = \sum_{m=n_1-\lfloor d/T \rfloor}^{n_2} g(n, m|d, \mathbf{w})[s(m) + z_1(m) - \hat{z}_1(m)], \quad n_1 - \lfloor d/T \rfloor \leq n \leq n_2. \quad (\text{A.29})$$

From the orthogonality principle [45], the following conditions are met by the above estimators

$$E[(z_1(n) - \hat{z}_1(n))z_2^*(m)] = 0 \quad (\text{A.30})$$

and

$$E[(s(n) - \hat{s}(n))(s^*(m) + z_1^*(m) - \hat{z}_1^*(m))] = 0, \quad (\text{A.31})$$

for $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$ and $n_1 - \lfloor d/T \rfloor \leq m \leq n_2$. From equations (A.25), (A.26) and (A.27), the following expected values are obtained

$$E[z_1(n)z_1^*(m)] = \begin{cases} \frac{N_o}{2}\rho_{w^{-1}}(n-m) & n_1 - \lfloor d/T \rfloor \leq n, m < n_1 \\ \frac{N_o}{8}[\delta(n-m) + \rho_{w^{-1}}(n-m)] & n_1 \leq n, m < n_2 - \lfloor d/T \rfloor \\ \frac{N_o}{2}\delta(n-m) & n_2 - \lfloor d/T \rfloor \leq n, m \leq n_2 \end{cases} \quad (\text{A.32})$$

$$E[z_1(n)z_2^*(m)] = \begin{cases} \frac{N_o}{8}[\delta(n-m) - \rho_{w^{-1}}(n-m)] & n_1 \leq n, m < n_2 - \lfloor d/T \rfloor \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.33})$$

$$E[z_2(n)z_2^*(m)] = \begin{cases} \frac{N_o}{8}[\delta(n-m) + \rho_{w^{-1}}(n-m)] & n_1 \leq n, m < n_2 - \lfloor d/T \rfloor \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.34}),$$

where $\rho_{w^{-1}}(k)$ is the deterministic autocorrelation of the inverse filter $w^{-1}(n)$ and is defined as [45]

$$\rho_{w^{-1}}(k) = \sum_i w^{-1}(i+k)w^{-1*}(i). \quad (\text{A.35})$$

A.2.1 The Estimator $f(n, m|d, h)$

Combining equations (A.28), (A.30), (A.33) and (A.34), it is found that $f(n, m|d, w)$ must satisfy

$$\sum_{k=n_1}^{n_2 - \lfloor d/T \rfloor - 1} f(n, k|d, w)[\delta(k-m) + \rho_{w^{-1}}(k-m)] = \delta(n-m) - \rho_{w^{-1}}(n-m), \quad (\text{A.36})$$

for $n_1 \leq n < n_2 - \lfloor d/T \rfloor$ and $n_1 \leq m < n_2 - \lfloor d/T \rfloor$. For a finite interval $[n_1 - \lfloor d/T \rfloor, n_2]$, equation (A.36) can be put in matrix form by defining the deterministic autocorrelation matrix

$$\mathbf{R}_\rho = \begin{bmatrix} \rho_{w^{-1}}(0) & \rho_{w^{-1}}(1) & \dots & \rho_{w^{-1}}(n_2 - n_1 - \lfloor d/T \rfloor - 1) \\ \rho_{w^{-1}}(-1) & \rho_{w^{-1}}(0) & \dots & \rho_{w^{-1}}(n_2 - n_1 - \lfloor d/T \rfloor - 2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{w^{-1}}(n_1 - n_2 + \lfloor d/T \rfloor) & \rho_{w^{-1}}(n_1 - n_2 + \lfloor d/T \rfloor + 2) & \dots & \rho_{w^{-1}}(0) \end{bmatrix} \quad (\text{A.37})$$

and the deterministic cross-correlation vector

$$\mathbf{P}_{\delta\rho}(n) = \begin{bmatrix} \delta(n - n_1) - \rho_{w^{-1}}(n - n_1) \\ \vdots \\ \delta(n - n_2 + \lfloor d/T \rfloor + 1) - \rho_{w^{-1}}(n - n_2 + \lfloor d/T \rfloor + 1) \end{bmatrix}. \quad (\text{A.38})$$

Define also the estimator vector

$$\mathbf{F}(n) = \begin{bmatrix} f(n, n_1 | d, \mathbf{w}) \\ \vdots \\ f(n, n_2 - \lfloor d/T \rfloor | d, \mathbf{w}) \end{bmatrix}. \quad (\text{A.39})$$

Equation (A.36) then becomes

$$(\mathbf{I} + \mathbf{R}_\rho)\mathbf{F}(n) = \mathbf{P}_{\delta\rho}(n) \quad (\text{A.40})$$

and its solution is

$$\mathbf{F}(n) = (\mathbf{I} + \mathbf{R}_\rho)^{-1} \mathbf{P}_{\delta\rho}(n). \quad (\text{A.41})$$

Note that the inverse in (A.41) exists since $w(n)$, and therefore $\rho_{w^{-1}}(n)$, is assumed invertible. Note also that the estimator impulse response is independent of the delay d . Defining the ij^{th} element of the matrix $(\mathbf{I} + \mathbf{R}_\rho)^{-1}$ as θ_{ij} , $f(n, m | d, \mathbf{w})$ can be expressed as

$$\begin{aligned} f(n, m | d, \mathbf{w}) &= \sum_{i=n_1}^{n_2 - \lfloor d/T \rfloor - 1} \theta_{im} [\delta(n-i) - \rho_{w^{-1}}(n-i)] \\ &= \theta_{nm} - \sum_{i=n_1}^{n_2 - \lfloor d/T \rfloor - 1} \theta_{im} \rho_{w^{-1}}(n-i). \end{aligned} \quad (\text{A.42})$$

A.2.2 The Estimator $g(n, m | d, \mathbf{w})$

From equations (A.29) and (A.31), the linear MMSE estimator $g(n, m | d, \mathbf{w})$ is the solution of

$$\begin{aligned} \phi_{ss}(n-m) &= E[\hat{s}(n)(s^*(m) + z_1^*(m) - \hat{z}_1^*(m))] \\ &= \sum_{k=n_1 - \lfloor d/T \rfloor}^{n_2} g(n, k | d, \mathbf{w}) E[(s(k) + z_1(k) - \hat{z}_1(k))(s^*(m) + z_1^*(m) - \hat{z}_1^*(m))] \\ &= \sum_{k=n_1 - \lfloor d/T \rfloor}^{n_2} g(n, k | d, \mathbf{w}) \phi_{ss}(k-m) \\ &+ \sum_{k=n_1 - \lfloor d/T \rfloor}^{n_2} g(n, k | d, \mathbf{w}) E[(z_1(k) - \hat{z}_1(k))(z_1^*(m) - \hat{z}_1^*(m))], \end{aligned} \quad (\text{A.43})$$

for $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$ and $n_1 - \lfloor d/T \rfloor \leq m \leq n_2$. Using equations (A.28), (A.29), (A.30), (A.32) and (A.33) in equation (A.43), $g(n, m|d, \mathbf{w})$ is the solution of

$$\begin{aligned}
\phi_{ss}(n-m) &= \sum_{k=n_1-\lfloor d/T \rfloor}^{n_2} g(n, k|d, \mathbf{w}) \phi_{ss}(k-m) \\
&+ \frac{N_o}{2} \sum_{k=n_1-\lfloor d/T \rfloor}^{n_1-1} g(n, k|d, \mathbf{w}) [\rho_{w-1}(k-m) - 1/4 f(k, m|\mathbf{w})] \\
&+ \frac{N_o}{8} \sum_{k=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, k|d, \mathbf{w}) [\delta(k-m) + \rho_{w-1}(k-m) - f(k, m|\mathbf{w})] \quad (\text{A.44}) \\
&+ \frac{N_o}{8} \sum_{k=n_1-\lfloor d/T \rfloor}^{n_2} \sum_{i=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, k|d, \mathbf{w}) f(k, i|\mathbf{w}) \rho_{w-1}(i-m) \\
&+ \frac{N_o}{2} \sum_{k=n_2-\lfloor d/T \rfloor}^{n_2} g(n, k|d, \mathbf{w}) [\delta(k-m) - 1/4 f(k, m|\mathbf{w})],
\end{aligned}$$

for $n_1 - \lfloor d/T \rfloor \leq n \leq n_2$ and $n_1 \leq m \leq n_2 - \lfloor d/T \rfloor - 1$. Note that all the terms in (A.44) involving $f(k, m|\mathbf{w})$ are zero for m outside $[n_1, n_2 - \lfloor d/T \rfloor - 1]$. Using equation (A.36) in equation (A.44) simplifies the result to

$$\begin{aligned}
\phi_{ss}(n-m) &= \\
&\sum_{k=n_1-\lfloor d/T \rfloor}^{n_1-1} g(n, k|d, \mathbf{w}) \left[\phi_{ss}(k-m) + \frac{N_o}{8} \{ \delta(k-m) + 3\rho_{w-1}(k-m) - 2f(k-m|\mathbf{w}) \} \right] \\
&+ \sum_{k=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, k|d, \mathbf{w}) \left[\phi_{ss}(k-m) + \frac{N_o}{4} \{ \delta(k-m) - f(k-m|\mathbf{w}) \} \right] \\
&+ \sum_{k=n_2-\lfloor d/T \rfloor}^{n_2} g(n, k|d, \mathbf{w}) \left[\phi_{ss}(k-m) + \frac{N_o}{8} \{ 5\delta(k-m) - \rho_{w-1}(k-m) - 2f(k-m|\mathbf{w}) \} \right]. \quad (\text{A.45})
\end{aligned}$$

A.2.3 Explicit Entries of $Q_2(n, m|d, \mathbf{w})$

From Figure A.1, the following relations, involving the entries $q_{ij}(n, m|d, \mathbf{w})$, are found

$$\begin{aligned}
\hat{s}_1(n) &= \sum_{m=n_1}^{n_2} q_{11}(n, m|d, \mathbf{w}) y_1(m) + q_{12}(n, m|d, \mathbf{w}) y_2(m) \\
\hat{s}_2(n) &= \sum_{m=n_1}^{n_2} q_{21}(n, m|d, \mathbf{w}) y_1(m) + q_{22}(n, m|d, \mathbf{w}) y_2(m). \quad (\text{A.46})
\end{aligned}$$

By solving equations (A.36) and (A.45), the linear MMSE estimator $f(n, m|\mathbf{w})$ and $g(n, m|d, \mathbf{w})$ are obtained. From Figure A.2, the following input-output relations are found

$$\begin{aligned}
\hat{s}_1(n) = & \sum_{m=n_1-\lfloor d/T \rfloor}^{n_1-1} g(n, m|d, \mathbf{w}) \mathcal{L}_{d, \mathbf{w}}^{-1}[y_2(m)] \\
& + 1/2 \sum_{m=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, m|d, \mathbf{w}) \left[y_1(m) - \sum_{\ell=n_1}^{n_2-\lfloor d/T \rfloor-1} f(m, \ell|\mathbf{w}) y_1(\ell) \right] \\
& + 1/2 \sum_{m=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, m|d, \mathbf{w}) \left[\mathcal{L}_{d, \mathbf{w}}^{-1}[y_2(m)] + \sum_{\ell=n_1}^{n_2-\lfloor d/T \rfloor-1} f(m, \ell|\mathbf{w}) \mathcal{L}_{d, \mathbf{w}}^{-1}[y_2(\ell)] \right] \\
& + \sum_{m=n_2-\lfloor d/T \rfloor}^{n_2} g(n, m|d, \mathbf{w}) y_1(m)
\end{aligned} \tag{A.47}$$

and

$$\hat{s}_2(n) = \mathcal{L}_{d, \mathbf{w}}[\hat{s}_1(n)]. \tag{A.48}$$

Define the following functions

$$b(n, m|d, \mathbf{w}) = \sum_{i=n_1}^{n_2-\lfloor d/T \rfloor-1} g(n, i|d, \mathbf{w}) f(i, m|\mathbf{w}) \tag{A.49}$$

$$p_1(n, m|d, \mathbf{w}) = \sum_{i=n_1}^{n_1+\lfloor d/T \rfloor-1} g(nT, iT-d|d, \mathbf{w}) w^{-1}(i-m) \tag{A.50}$$

$$p_2(n, m|d, \mathbf{w}) = \sum_{i=n_1+\lfloor d/T \rfloor+1}^{n_2} g(nT, iT-d|d, \mathbf{w}) w^{-1}(i-m) \tag{A.51}$$

$$a(n, m|d, \mathbf{w}) = \sum_{i=n_1+\lfloor d/T \rfloor}^{n_2-1} b(nT, iT-d|d, \mathbf{w}) w^{-1}(i-m). \tag{A.52}$$

Using equations (A.26),(A.49) to (A.52) and performing a change of variables, equation (A.47) becomes

$$\begin{aligned}
\hat{s}_1(n) = & 1/2 \sum_{m=n_1}^{n_2-\lfloor d/T \rfloor-1} [g(n, m|d, \mathbf{w}) - b(n, m|d, \mathbf{w})] y_1(m) \\
& + \sum_{m=n_2-\lfloor d/T \rfloor}^{n_2} g(n, m|d, \mathbf{w}) y_1(m) \\
& + \sum_{m=n_1}^{n_1+\lfloor d/T \rfloor-1} p_1(n, m|d, \mathbf{w}) y_2(m) \\
& + 1/2 \sum_{m=n_1+\lfloor d/T \rfloor}^{n_2-1} [p_2(n, m|d, \mathbf{w}) + a(n, m|d, \mathbf{w})] y_2(m).
\end{aligned} \tag{A.53}$$

Comparing equations (A.46) and (A.53) and using (A.48), the explicit forms for the $q_{ij}(n, m|d, \mathbf{w})$'s are

$$q_{11}(n, m|d, \mathbf{w}) = \begin{cases} 1/2[g(n, m|d, \mathbf{w}) - b(n, m|d, \mathbf{w})] & n_1 \leq m \leq n_2 - \lfloor d/T \rfloor - 1 \\ g(n, m|d, \mathbf{w}) & n_2 - \lfloor d/T \rfloor \leq m \leq n_2 \end{cases} \quad (\text{A.54})$$

$$q_{12}(n, m|d, \mathbf{w}) = \begin{cases} p_1(n, m|d, \mathbf{w}) & n_1 \leq m \leq n_1 + \lfloor d/T \rfloor - 1 \\ 1/2[p_2(n, m|d, \mathbf{w}) + a(n, m|d, \mathbf{w})] & n_1 + \lfloor d/T \rfloor \leq m \leq n_2 - 1 \end{cases} \quad (\text{A.55})$$

$$q_{21}(n, m|d, \mathbf{w}) = \mathcal{L}_{d, \mathbf{w}}[q_{11}(n, m|d, \mathbf{w})] \quad (\text{A.56})$$

$$q_{22}(n, m|d, \mathbf{w}) = \mathcal{L}_{d, \mathbf{w}}[q_{12}(n, m|d, \mathbf{w})] \quad (\text{A.57})$$

and $n_1 \leq n \leq n_2$.

A.3 The function $\ell_Y(d, \mathbf{w})$ for a Long Observation Interval

The function $\ell_Y(d, \mathbf{w})$, when $n_1 \rightarrow -\infty$ and $n_2 \rightarrow \infty$, is computed in this section by using the results derived in Section A.2. Because the function $\rho_{w^{-1}}(n)$ is invertible (the reference impulse response $w(n)$ is assumed invertible), the time-invariant functions $f(n|\mathbf{w})$ and $g(n|\mathbf{w})$ are solutions of equations (A.36) and (A.45) respectively. Then, using equations (A.49) to (A.52) in equations (A.54) to (A.57), and neglecting the terms involving $\lfloor d/T \rfloor$, (because of the large observation assumption) the entries of the matrix impulse response $Q_2(n|d, \mathbf{w})$ are also time-invariant and are given by

$$\begin{aligned} q_{11}(n|\mathbf{w}) &= 1/2[\delta(n) - f(n|\mathbf{w})] \otimes g(n|\mathbf{w}) \\ q_{12}(n|d, \mathbf{w}) &= 1/2[\delta(n) + f(n|\mathbf{w})] \otimes g(nT + d|\mathbf{w}) \otimes w^{-1}(n) \\ q_{21}(n|d, \mathbf{w}) &= 1/2[\delta(n) - f(n|\mathbf{w})] \otimes g(nT - d|\mathbf{w}) \otimes w(n) \\ q_{22}(n|\mathbf{w}) &= 1/2[\delta(n) + f(n|\mathbf{w})] \otimes g(n|\mathbf{w}). \end{aligned} \quad (\text{A.58})$$

From equation (A.20), and from the above definitions of the matrix entries, when $n_1 \rightarrow -\infty$ and $n_2 \rightarrow \infty$, the likelihood $\ell_Y(d, \mathbf{w})$ is given by

$$\begin{aligned} \ell_Y(d, \mathbf{w}) &= 1/2N_o \sum_n [(\delta(n) + f(n|\mathbf{w})) \otimes g(nT + d|\mathbf{w}) \otimes w^{-1}(n) \otimes y_2(n)] y_1^*(n) \\ &+ 1/2N_o \sum_n [(\delta(n) - f(n|\mathbf{w})) \otimes g(nT - d|\mathbf{w}) \otimes w(n) \otimes y_1(n)] y_2^*(n) \\ &+ 1/2N_o \sum_n [(\delta(n) - f(n|\mathbf{w})) \otimes g(n|\mathbf{w}) \otimes y_1(n)] y_1^*(n) \\ &+ 1/2N_o \sum_n [(\delta(n) + f(n|\mathbf{w})) \otimes g(n|\mathbf{w}) \otimes y_2(n)] y_2^*(n). \end{aligned} \quad (\text{A.59})$$

Note that the first two terms of equation (A.59) can be written as

$$\begin{aligned} \ell_{Y_1}(d, \mathbf{w}) &= 1/2N_o \sum_n [(\delta(n) + f(n|\mathbf{w})) \otimes g(n|\mathbf{w}) \otimes w^{-1}(n) \otimes y_2(nT + d)] y_1^*(n) \\ &+ 1/2N_o \sum_n [(\delta(n) - f(n|\mathbf{w})) \otimes g(n|\mathbf{w}) \otimes w(n) \otimes y_1(nT - d)] y_2^*(n). \end{aligned} \quad (\text{A.60})$$

Define

$$w_1(n|\mathbf{w}) = [\delta(n) - f(n|\mathbf{w})] \otimes g(n|\mathbf{w}) \otimes w(n) \quad (\text{A.61})$$

and

$$\tilde{w}(n|\mathbf{w}) = [\delta(n) + f(n|\mathbf{w})] \otimes g(n|\mathbf{w}) \otimes w^{-1}(n). \quad (\text{A.62})$$

Use definitions (A.61) and (A.62) and equation (A.60) in equation (A.59) in order to get

$$\begin{aligned} \ell_Y(d, \mathbf{w}) &= 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes y_2(nT + d)] y_1^*(n) \\ &+ 1/2N_o \sum_n [w_1(n|\mathbf{w}) \otimes y_1(nT - d)] y_2^*(n) \\ &+ 1/2N_o \sum_n [w_1(n|\mathbf{w}) \otimes w^{-1}(n) \otimes y_1(n)] y_1^*(n) \\ &+ 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes w(n) \otimes y_2(n)] y_2^*(n). \end{aligned} \quad (\text{A.63})$$

When $n_1 \rightarrow -\infty$ and $n_2 \rightarrow \infty$, equations (A.36) and (A.45) become respectively, in the frequency domain,

$$F(e^{j\omega})[1 + |W_{\text{inv}}(e^{j\omega})|^2] = 1 - |W_{\text{inv}}(e^{j\omega})|^2 \quad (\text{A.64})$$

and

$$\Phi_{ss}(e^{j\omega}) = G(e^{j\omega}) \left[\Phi_{ss}(e^{j\omega}) + \frac{N_o}{4} \{1 - F(e^{j\omega})\} \right]. \quad (\text{A.65})$$

Note that $W_{\text{inv}}(e^{j\omega})$ is the Fourier transform of $w^{-1}(n)$ and is defined as $1/W(e^{j\omega})$.

From equation (A.64), the linear MMSE estimator $f(n|\mathbf{w})$ has the following frequency response

$$F(e^{j\omega}) = \frac{|W(e^{j\omega})|^2 - 1}{|W(e^{j\omega})|^2 + 1}. \quad (\text{A.66})$$

From equations (A.65) and (A.66), the linear estimator $g(n|\mathbf{w})$ has the following frequency response

$$G(e^{j\omega}) = \frac{\Phi_{ss}(e^{j\omega})(|W(e^{j\omega})|^2 + 1)}{\Phi_{ss}(e^{j\omega})(|W(e^{j\omega})|^2 + 1) + N_o/2}. \quad (\text{A.67})$$

The impulse responses $w_1(n|\mathbf{w})$ and $\tilde{w}(n|\mathbf{w})$ can then be expressed as

$$\begin{aligned} \tilde{w}(n|\mathbf{w}) &= \mathbf{F}^{-1} \left[\frac{2G(e^{j\omega})W^*(e^{j\omega})}{|W(e^{j\omega})|^2 + 1} \right] \\ w_1(n|\mathbf{w}) &= \tilde{w}(n|\mathbf{w}) \otimes c_w(n). \end{aligned} \quad (\text{A.68})$$

where $\mathbf{F}^{-1}[\cdot]$ is the inverse Fourier transform operator and $c_w(n)$ is defined as

$$c_w(n) = w(n) \otimes w(n).$$

Note that

$$\begin{aligned} \sum_n [\tilde{w}(n|\mathbf{w}) \otimes y_2(nT + d)] y_1^*(n) &= \sum_n \sum_k \tilde{w}(k|\mathbf{w}) y_2(nT - kT + d) y_1^*(n) \\ &= \sum_n \sum_\ell \tilde{w}(nT - \ell T + d|\mathbf{w}) y_2(\ell) y_1^*(n) \\ &= \sum_\ell \sum_j \tilde{w}(j - \ell|\mathbf{w}) y_1^*(jT - d) y_2(\ell) \\ &= \sum_n [\tilde{w}(-n|\mathbf{w}) \otimes y_1^*(nT - d)] y_2(n). \end{aligned} \tag{A.69}$$

From equations (A.68) and (A.69), equation (A.63) can then be written as

$$\begin{aligned} \ell_Y(d, \mathbf{w}) &= 1/2N_o \sum_n [\tilde{w}(-n|\mathbf{w}) \otimes y_1^*(nT - d)] y_2(n) \\ &\quad + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes c_w(n) \otimes y_1(nT - d)] y_2^*(n) \\ &\quad + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes w(n) \otimes y_1(n)] y_1^*(n) \\ &\quad + 1/2N_o \sum_n [\tilde{w}(n|\mathbf{w}) \otimes w(n) \otimes y_2(n)] y_2^*(n). \end{aligned} \tag{A.70}$$

Appendix B. The Ordinary Differential Equation (ODE) Method

A heuristic discussion on the development of the ordinary differential equations associated with equations (4.9) and (4.15) of section 4.2 is presented in this appendix and can be found in [53] or in [4].

Assume a recursive parameter vector estimation method of the form of equation (4.9), but with a *scalar* gain sequence $\gamma(n)$, i.e.

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \gamma(n)\mathbf{R}^{-1}(n)\boldsymbol{\psi}(n)\epsilon(n), \quad (\text{B.1})$$

with

$$\mathbf{R}(n+1) = \mathbf{R}(n) + \gamma(n+1)[\boldsymbol{\psi}(n+1)\boldsymbol{\psi}^T(n+1) - \mathbf{R}(n)]. \quad (\text{B.2})$$

Assume that the parameter vector $\boldsymbol{\theta}(n)$ approaches $\boldsymbol{\theta}(n-1)$ asymptotically (subject to regularity conditions such as stability, stationarity, etc.). Express $\boldsymbol{\theta}(n+N)$ as

$$\boldsymbol{\theta}(n+N) = \boldsymbol{\theta}(n) + \sum_{k=n}^{n+N-1} \gamma(k)\mathbf{R}^{-1}(k)\boldsymbol{\psi}(k)\epsilon(k), \quad (\text{B.3})$$

where equation (B.1) is used. Suppose that $\mathbf{R}^{-1}(k)$ in equation (B.3) is fixed to be $\mathbf{R}^{-1}(n)$ for the interval N . Using the law of large numbers, the summation of the terms $\boldsymbol{\psi}(k)\epsilon(k)$ can be approximated by $E[\boldsymbol{\psi}(k)\epsilon(k)]$ and equation (B.3) can be approximately written as

$$\boldsymbol{\theta}(n+N) \approx \boldsymbol{\theta}(n) + \mathbf{R}^{-1}(n)E[\boldsymbol{\psi}(k)\epsilon(k)] \sum_{k=n}^{n+N-1} \gamma(k), \quad (\text{B.4})$$

where the expected value $E[\boldsymbol{\psi}(k)\epsilon(k)]$ is defined as $f(\boldsymbol{\theta}(n))$. Define the compressed time scales:

$$\begin{aligned} \tau &= \sum_{k=1}^n \gamma(k) \\ \Delta\tau &= \sum_{k=n}^{n+N-1} \gamma(k). \end{aligned} \quad (\text{B.5})$$

Changing the time scale from n to τ and mapping $\boldsymbol{\theta}(n)$ into $\boldsymbol{\theta}_D(\tau)$, equation (B.4) becomes

$$\boldsymbol{\theta}_D(\tau + \Delta\tau) \approx \boldsymbol{\theta}_D(\tau) + \Delta\tau\mathbf{R}^{-1}(\tau)f(\boldsymbol{\theta}_D(\tau)). \quad (\text{B.6})$$

Asymptotically, when $\Delta\tau$ becomes small, (B.6) reduces to

$$\frac{d\boldsymbol{\theta}_D(\tau)}{d\tau} \approx \mathbf{R}^{-1}(\tau)f(\boldsymbol{\theta}_D(\tau)), \quad (\text{B.7})$$

which is approximately the first equation associated with (B.1) and (B.2). The second equation, given by

$$\frac{d\mathbf{R}(\tau)}{d\tau} = G[\boldsymbol{\theta}_D(\tau)] - \mathbf{R}(\tau) \quad (\text{B.8})$$

with

$$G[\boldsymbol{\theta}_D(\tau)] = E[\boldsymbol{\psi}(n)\boldsymbol{\psi}^T(n)], \quad (\text{B.9})$$

is heuristically obtained in a similar way. Note that the case of a matrix gain sequence $\boldsymbol{\gamma}(n)$ is comprised in the above derivations when $\boldsymbol{\gamma}(n) = \gamma(n)\mathbf{I}$.

Appendix C. Cross-Correlation of Differentiated Random Processes

The relations derived in this appendix are obtained using the theory of linear systems with stochastic inputs. The results presented below are often used in the analyses and follow the examples given in reference [45], pp. 237-239.

Consider two *stationary* complex random processes $x(t)$ and $y(t)$. Their cross-correlation $\phi_{xy}(\tau)$ is defined as

$$\phi_{xy}(\tau) = E[x(t_1)y^*(t_2)] \quad (\text{C.1})$$

where τ is defined as

$$\tau = t_1 - t_2. \quad (\text{C.2})$$

It is assumed that the two random processes are delayed by the same delay d , i.e.

$$x(t_1) = x(t - d)$$

$$y(t_2) = y(t - \tau - d).$$

In the following sections, the derivative with respect to the delay d and with respect to τ are denoted as follows

$$\begin{aligned} \frac{\partial x(\cdot)}{\partial d} &= \dot{x}(\cdot) \\ \frac{\partial^2 x(\cdot)}{\partial d^2} &= \ddot{x}(\cdot) \\ \frac{\partial x(\cdot)}{\partial \tau} &= x'(\cdot) \\ \frac{\partial^2 x(\cdot)}{\partial \tau^2} &= x''(\cdot). \end{aligned} \quad (\text{C.3})$$

C.1 Cross-correlation of $\dot{x}(t)$ and $y(t)$

Because of the linearity of the differentiation operator, the following is true

$$\begin{aligned} E[\dot{x}(t_1)y^*(t_2)] &= E\left[\frac{\partial x(t_1)}{\partial d}y^*(t_2)\right] \\ &= E\left[\frac{\partial t_1}{\partial d}\frac{\partial x(t_1)}{\partial t_1}y^*(t_2)\right] \\ &= \frac{\partial t_1}{\partial d}\frac{\partial E[x(t_1)y^*(t_2)]}{\partial t_1}. \end{aligned} \quad (\text{C.4})$$

Noting that

$$\frac{\partial t_1}{\partial d} = \frac{\partial t_2}{\partial d} = -1,$$

equation (C.4) becomes

$$E[\dot{x}(t_1)y^*(t_2)] = -\frac{\partial \tau}{\partial t_1}\frac{\partial \phi_{xy}(\tau)}{\partial \tau}. \quad (\text{C.5})$$

Therefore

$$E[\dot{x}(t_1)y^*(t_2)] = -\phi'_{xy}(\tau). \quad (\text{C.6})$$

C.2 Cross-correlation of $x(t)$ and $\dot{y}(t)$

Using the same type of development as above, we have

$$\begin{aligned} E[x(t_1)\dot{y}^*(t_2)] &= E\left[x(t_1)\frac{\partial y^*(t_2)}{\partial d}\right] \\ &= E\left[\frac{\partial t_2}{\partial d}x(t_1)\frac{\partial y^*(t_2)}{\partial t_2}\right] \\ &= \frac{\partial t_2}{\partial d}\frac{\partial \tau}{\partial t_2}\frac{\partial \phi_{xy}(\tau)}{\partial \tau}. \end{aligned}$$

Then

$$E[x(t_1)\dot{y}^*(t_2)] = \phi'_{xy}(\tau). \quad (\text{C.7})$$

C.3 Cross-correlation of $\dot{x}(t)$ and $\dot{y}(t)$

Using the results of the last two sections

$$\begin{aligned} E[\dot{x}(t_1)\dot{y}^*(t_2)] &= \frac{\partial E[\dot{x}(t_1)\dot{y}^*(t_2)]}{\partial \tau} \\ &= \frac{\partial \phi_{\dot{x}\dot{y}}(\tau)}{\partial \tau} \\ &= -\frac{\partial^2 \phi_{xy}(\tau)}{\partial \tau^2} \end{aligned}$$

the desired cross-correlation is

$$E[\dot{x}(t_1)\dot{y}^*(t_2)] = -\phi''_{xy}(\tau). \quad (\text{C.8})$$

C.4 Cross-correlation of $\ddot{x}(t)$ and $y(t)$

From the double application of results (C.6), we have

$$\begin{aligned} E[\ddot{x}(t_1)y^*(t_2)] &= -\phi'_{\dot{x}y}(\tau) \\ &= -\frac{\partial(-\phi'_{xy}(\tau))}{\partial d} \\ &= \phi''_{xy}(\tau). \end{aligned} \quad (\text{C.9})$$

C.5 Cross-correlation of $x(t)$ and $\ddot{y}(t)$

From the double application of results (C.7), we have

$$\begin{aligned} E[x(t_1)\ddot{y}^*(t_2)] &= \phi'_{x\dot{y}}(\tau) \\ &= \phi''_{xy}(\tau). \end{aligned} \quad (\text{C.10})$$

Appendix D. Some Expected Values For a Type I Adaptive System

The expected values necessary in the computations of Section 4.3.1 are derived in this appendix. The results of equations (4.34) to (4.38) and those of appendix C are used in the following derivations.

D.1 Expected Value of G_n

The quantity G_n is defined in (4.45) and its expected value is

$$\begin{aligned} E[G_n] &= E[\dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D)] \\ &= E[\dot{y}^2(n, D)] - E[e(n, D)\ddot{y}(n, D)]. \end{aligned} \quad (D.1)$$

From (4.34) to (4.38) and appendix C, we have

$$\begin{aligned} E[\dot{y}^2(n, D)] &= E[(\partial\hat{r}(n)/\partial d_n + \dot{\chi}(n, D))^2] \\ &= E[(\partial\hat{r}(n)/\partial d_n)^2] + E[\dot{\chi}^2(n, D)] \\ &= -\phi''_{\hat{r}\hat{r}}(0) + E[\dot{\chi}^2(n, D)]. \end{aligned} \quad (D.2)$$

The quantity $E[\dot{\chi}^2(n, D)]$ is expressed as

$$\begin{aligned} E[\dot{\chi}^2(n, D)] &= E[\eta_n^T \dot{\mathbf{u}}(nT - D) \eta_n^T \dot{\mathbf{u}}(nT - D)] \\ &= E\left[\sum_i \sum_j \eta_i(n) \dot{u}_i(nT - D) \eta_j(n) \dot{u}_j(nT - D)\right] \\ &= \sum_i \sum_j E[\eta_i(n) \dot{u}_i(nT - D) \eta_j(n) \dot{u}_j(nT - D)]. \end{aligned} \quad (D.3)$$

Since the input vector and noise vector components are assumed to be formed of Gaussian random variables, we have [66]

$$\begin{aligned} E[\eta_i(n) \dot{u}_i(nT - D) \eta_j(n) \dot{u}_j(nT - D)] &= E[\eta_i(n) \dot{u}_i(nT - D)] E[\eta_j(n) \dot{u}_j(nT - D)] \\ &\quad + E[\dot{u}_i(nT - D) \eta_j(n)] E[\eta_i(n) \dot{u}_j(nT - D)] \\ &\quad + E[\dot{u}_i(nT - D) \dot{u}_j(nT - D)] E[\eta_i(n) \eta_j(n)]. \end{aligned} \quad (D.4)$$

Every correlation of a noise vector component with an input sample is zero and, from assumption 5 of Section 4.3,

$$E[\eta_i(n) \eta_j(n)] = 0 \quad \text{for } i \neq j.$$

Therefore, equation (D.3) simplifies to

$$\begin{aligned} E[\dot{\chi}^2(n, D)] &= \sum_i E[\dot{u}_i^2(nT - D)] E[\eta_i^2(n)] \\ &= -\phi''_{uu}(0) \sum_i E[\eta_i^2(n)]. \end{aligned} \quad (D.5)$$

The sum of the variances of the noise vector components is equal the trace of the correlation matrix of $\boldsymbol{\eta}_n$, defined as

$$\mathbf{K}_\eta = E[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T], \quad (\text{D.6})$$

and (D.5) can be written as

$$E[\dot{\chi}^2(n, D)] = -\phi''_{uu}(0) \text{tr}[\mathbf{K}_\eta]. \quad (\text{D.7})$$

Then, equation (D.2) is

$$E[\dot{y}^2(n, D)] = -\phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0) \text{tr}[\mathbf{K}_\eta]. \quad (\text{D.8})$$

The second component of $E[G_n]$ is

$$\begin{aligned} E[e(n, D)\ddot{y}(n, D)] &= E[(r(n) - y(n, D))\ddot{y}(n, D)] \\ &= E[r(n)(\partial^2 \hat{r}(n)/\partial d_n^2 + \ddot{\chi}(n, D))] - E[y(n, D)\ddot{y}(n, D)] \\ &= \phi''_{r\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0) \text{tr}[\mathbf{K}_\eta] \\ &\approx \phi''_{\hat{r}\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0) \text{tr}[\mathbf{K}_\eta] \\ &\approx -\phi''_{uu}(0) \text{tr}[\mathbf{K}_\eta], \end{aligned} \quad (\text{D.9})$$

where the last two approximations come from the high signal-to-noise ratios assumption.

From (D.1), $E[G_n]$ is then

$$\begin{aligned} E[G_n] &= -\phi''_{r\hat{r}}(0) \\ &\approx -\phi''_{\hat{r}\hat{r}}(0), \end{aligned} \quad (\text{D.10})$$

for high signal-to-noise ratios.

D.2 Expected Value of $(1 - 2\alpha G_n)N_n$

From equations (4.44) and (4.45), the expected value is written as

$$\begin{aligned} E[(1 - 2\alpha G_n)N_n] &= E[N_n] - 2\alpha E[G_n N_n] \\ &= 4\alpha E[(\dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D))(e(n, D)\dot{y}(n, D))] \\ &= 4\alpha (E[\dot{y}^3(n, D)e(n, D)] - E[\dot{y}(n, D)\ddot{y}(n, D)e^2(n, D)]). \end{aligned} \quad (\text{D.11})$$

All the random variables involved in (D.11) are assumed zero-mean Gaussian and from the fact that $E[N_n] = -2E[e(n, D)\dot{y}(n, D)] = 0$, we have

$$\begin{aligned} E[\dot{y}^3(n, D)e(n, D)] &= 3E[\dot{y}^2(n, D)]E[\dot{y}(n, D)e(n, D)] \\ &= (-3/2)E[\dot{y}^2(n, D)]E[N_n] \\ &= 0 \end{aligned} \quad (\text{D.12})$$

and

$$\begin{aligned}
E[\dot{y}(n, D)\ddot{y}(n, D)e^2(n, D)] &= E[\dot{y}(n, D)\ddot{y}(n, D)]E[e^2(n, D)] \\
&\quad + 2E[\ddot{y}(n, D)e(n, D)]E[\dot{y}(n, D)e(n, D)] \\
&= -(\phi_{\hat{r}\hat{r}}^{(3)}(0) + \phi_{\chi\chi}^{(3)}(0))E[e^2(n, D)] \\
&= 0.
\end{aligned} \tag{D.13}$$

This last result follows from the autocorrelation property that states [74]

$$\phi_{uu}(-\tau) = \phi_{uu}(\tau) \leq \phi_{uu}(0), \quad \tau \neq 0, \tag{D.14}$$

which forces the first and third derivatives of the autocorrelation to be zero at $\tau = 0$.

The final result is then

$$E[(1 - 2\alpha G_n)N_n] = 0. \tag{D.15}$$

D.3 Expected Value of G_n^2

From equation (4.45), this expected value is written as

$$\begin{aligned}
E[G_n^2] &= E[(\dot{y}^2(n, D) - e(n, D)\ddot{y}(n, D))^2] \\
&= E[\dot{y}^4(n, D)] - 2E[\dot{y}^2(n, D)e(n, D)\ddot{y}(n, D)] + E[e^2(n, D)\ddot{y}^2(n, D)].
\end{aligned} \tag{D.16}$$

From equation (4.34), the first term of (D.16) is expressed as

$$\begin{aligned}
E[\dot{y}^4(n, D)] &= E[(\partial\hat{r}/\partial d_n + \dot{\chi}(n, D))^4] \\
&= E[(\partial\hat{r}/\partial d_n)^4] + 4E[(\partial\hat{r}/\partial d_n)^3\dot{\chi}(n, D)] + 6E[(\partial\hat{r}/\partial d_n)^2\dot{\chi}^2(n, D)] \\
&\quad + 4E[(\partial\hat{r}/\partial d_n)\dot{\chi}^3(n, D)] + E[\dot{\chi}^4(n, D)].
\end{aligned} \tag{D.17}$$

Since $\partial\hat{r}/\partial d_n$ and $\dot{\chi}(n, D)$ are assumed to be zero-mean independent Gaussian random variables, the second and fourth terms on the right of (D.17) are zero. The first term is [45]

$$\begin{aligned}
E[(\partial\hat{r}/\partial d_n)^4] &= 3(E[(\partial\hat{r}/\partial d_n)^2])^2 \\
&= 3(\phi_{\hat{r}\hat{r}}''(0))^2.
\end{aligned} \tag{D.18}$$

The third term is

$$\begin{aligned}
6E[(\partial\hat{r}/\partial d_n)^2\dot{\chi}^2(n, D)] &= 6E[(\partial\hat{r}/\partial d_n)^2]E[\dot{\chi}^2(n, D)] \\
&= 6\phi_{\hat{r}\hat{r}}''(0)\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta]
\end{aligned} \tag{D.19}$$

where the result of (D.7) was used.

From a development analog to equations (D.3) to (D.7) and assuming that $\chi(n, D)$ is Gaussian, the fifth term of (D.17) is found to be

$$\begin{aligned}
E[\dot{\chi}^4(n, D)] &= 3(E[\dot{\chi}^2(n, D)])^2 \\
&= 3(\phi_{\dot{u}\dot{u}}(0) \sum_i E[\eta_i^2(n)])^2 \\
&= 3(\phi_{uu}''(0)\text{tr}[\mathbf{K}_\eta])^2.
\end{aligned} \tag{D.20}$$

Collecting the results of (D.18), (D.19) and (D.20), we have

$$E[\dot{y}^4(n, D)] = 3(\phi''_{\hat{r}\hat{r}}(0))^2 + 6\phi''_{\hat{r}\hat{r}}(0)\phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta] + 3(\phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta])^2. \quad (\text{D.21})$$

The expected value $E[\dot{y}^2(n, D)e(n, D)\ddot{y}(n, D)]$ in the second term of (D.16) is computed as follows

$$\begin{aligned} E[\dot{y}^2(n, D)e(n, D)\ddot{y}(n, D)] &= E[\dot{y}^2(n, D)]E[e(n, D)\ddot{y}(n, D)] \\ &\quad + 2E[\dot{y}(n, D)e(n, D)]E[\dot{y}(n, D)\ddot{y}(n, D)] \\ &= -(\phi''_{\hat{r}\hat{r}}(0) + \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi''_{\hat{r}\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta]), \end{aligned} \quad (\text{D.22})$$

where equations (D.8) and (D.9) were used. The third term of equation (D.16) is computed as follows

$$\begin{aligned} E[e^2(n, D)\dot{y}^2(n, D)] &= E[e^2(n, D)]E[\dot{y}^2(n, D)] + 2E^2[e(n, D)\dot{y}(n, D)] \\ &= E[(r(n) - y(n, D))^2]E[(\partial^2\hat{r}/\partial d_n^2 + \ddot{\chi}(n, D))^2] \\ &\quad + 2(\phi''_{\hat{r}\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta])^2 \\ &= (\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta]) \\ &\quad + 2(\phi''_{\hat{r}\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta])^2. \end{aligned} \quad (\text{D.23})$$

Collecting equations (D.21), (D.22) and (D.23), the final result is

$$\begin{aligned} E[G_n^2] &= 3(\phi''_{\hat{r}\hat{r}}(0))^2 + 6\phi''_{\hat{r}\hat{r}}(0)\phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta] + 3(\phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta])^2 \\ &\quad + (\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta]) \\ &\quad + 2\phi''_{\hat{r}\hat{r}}(0)(\phi''_{\hat{r}\hat{r}}(0) - \phi''_{\hat{r}\hat{r}}(0) - \phi''_{uu}(0)\text{tr}[\mathbf{K}_\eta]). \end{aligned} \quad (\text{D.24})$$

D.4 Expected Value of N_n^2

Using equation (4.44) and the results (D.23), (D.8), this expected value is

$$\begin{aligned} E[N_n^2] &= 4E[e^2(n, D)\dot{y}^2(n, D)] \\ &= 4E[e^2(n, D)]E[\dot{y}^2(n, D)] + 8E^2[N_n] \\ &= -4(\phi_{rr}(0) - \phi_{\hat{r}\hat{r}}(0) + \phi_{uu}(0)\text{tr}[\mathbf{K}_\eta])(\phi_{\hat{r}\hat{r}}^{(4)}(0) + \phi_{uu}^{(4)}(0)\text{tr}[\mathbf{K}_\eta]). \end{aligned} \quad (\text{D.25})$$

Appendix E. Shift Invariance Properties and Common Recursions in the LS algorithm: Type II-DRB

E.1 Shift Invariance Properties in the LS algorithm: Type II-DRB

Based on the definitions of Subsection 5.2.1, the following shift invariance properties can be established

$$\mathbf{u}_M(i) = \begin{bmatrix} \mathbf{u}_{M-1}(i) \\ u(i-M+1) \end{bmatrix} = \begin{bmatrix} u(i) \\ \mathbf{u}_{M-1}(i-1) \end{bmatrix}. \quad (\text{E.1})$$

Using (E.1) in (5.19)

$$\begin{aligned} \boldsymbol{\theta}_M^\ell(n) &= \begin{bmatrix} \sum_{i=1}^n \beta^{n-i} \mathbf{u}_{M-1}(i) r^*(i+\ell) \\ \sum_{i=1}^n \beta^{n-i} u(i-M+1) r^*(i+\ell) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\theta}_{M-1}^\ell(n) \\ \boldsymbol{\theta}_M^\ell(n) \end{bmatrix}, \end{aligned} \quad (\text{E.2})$$

where

$$\boldsymbol{\theta}_M^\ell(n) = \sum_{i=1}^n \beta^{n-i} u(i-M+1) r^*(i+\ell). \quad (\text{E.3})$$

Also,

$$\begin{aligned} \boldsymbol{\theta}_M^\ell(n) &= \begin{bmatrix} \sum_{i=1}^n \beta^{n-i} u(i) r^*(i+\ell) \\ \sum_{i=1}^n \beta^{n-i} \mathbf{u}_{M-1}(i-1) r^*(i+\ell) \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\boldsymbol{\theta}}_M^\ell(n) \\ \boldsymbol{\theta}_{M-1}^{\ell+1}(n-1) \end{bmatrix}, \end{aligned} \quad (\text{E.4})$$

where

$$\tilde{\boldsymbol{\theta}}_M^\ell(n) = \sum_{i=1}^n \beta^{n-i} u(i) r^*(i+\ell) \quad (\text{E.5})$$

and

$$\sum_{i=1}^n \beta^{n-i} \mathbf{u}_{M-1}(i-1) r^*(i+\ell)$$

$$\begin{aligned}
&= \sum_{i=0}^{n-1} \beta^{n-1-i} \mathbf{u}_{M-1}(i) r^*(i + \ell + 1) \\
&= \sum_{i=1}^{n-1} \beta^{n-1-i} \mathbf{u}_{M-1}(i) r^*(i + \ell + 1) + \beta^{n-1} \mathbf{u}_{M-1}(0) r^*(\ell + 1) \\
&= \theta_{M-1}^{\ell+1} (n-1),
\end{aligned} \tag{E.6}$$

since $\mathbf{u}_{M-1}(0) = \mathbf{0}$ in the prewindowed method. The following shift invariances can also be established [7]

$$\Phi_M(n) = \begin{bmatrix} \Phi_{M-1}(n) & \mathbf{r}_{M-1}^b(n) \\ \mathbf{r}_{M-1}^{bH}(n) & r_{M-1}^{b0}(n) \end{bmatrix} \tag{E.7}$$

or

$$\Phi_M(n) = \begin{bmatrix} r_{M-1}^{f0}(n) & \mathbf{r}_{M-1}^{fH}(n) \\ \mathbf{r}_{M-1}^f(n) & \Phi_{M-1}(n-1) \end{bmatrix}, \tag{E.8}$$

where

$$\mathbf{r}_{M-1}^b(n) = \sum_{i=1}^n \beta^{n-i} \mathbf{u}_{M-1}(i) u^*(i - M + 1) \tag{E.9}$$

$$\mathbf{r}_{M-1}^f(n) = \sum_{i=1}^n \beta^{n-i} \mathbf{u}_{M-1}(i-1) u^*(i) \tag{E.10}$$

$$r_{M-1}^{b0}(n) = \sum_{i=1}^n \beta^{n-i} |u(i - M + 1)|^2 \tag{E.11}$$

$$r_{M-1}^{f0}(n) = \sum_{i=1}^n \beta^{n-i} |u(i)|^2. \tag{E.12}$$

E.2 Common Recursions

The two following recursions are easily derived

$$\Phi_M(n) = \beta \Phi_M(n-1) + \mathbf{u}_M(n) \mathbf{u}_M^H(n) \tag{E.13}$$

$$\theta_M^\ell(n) = \beta \theta_M^\ell(n-1) + \mathbf{u}_M(n) r^*(n + \ell). \tag{E.14}$$

Using the matrix inversion lemma, the following recursion is obtained [7]

$$\Phi_M^{-1}(n) = \beta^{-1} \Phi_M^{-1}(n-1) - \frac{\beta^{-2} \Phi_M^{-1}(n-1) \mathbf{u}_M(n) \mathbf{u}_M^H(n) \Phi_M^{-1}(n-1)}{1 + \beta^{-1} \mathbf{u}_M^H(n) \Phi_M^{-1}(n-1) \mathbf{u}_M(n)}. \tag{E.15}$$

Define

$$\mathbf{g}_M(n) = \Phi_M^{-1}(n-1) \mathbf{u}_M(n) \tag{E.16}$$

and

$$\gamma_M(n) = 1 + \beta^{-1} \mathbf{g}_M^H(n) \mathbf{u}_M(n). \quad (\text{E.17})$$

Then (E.15) becomes

$$\Phi_M^{-1}(n) = \beta^{-1} \Phi_M^{-1}(n-1) - \beta^{-2} \frac{\mathbf{g}_M(n) \mathbf{g}_M^H(n)}{\gamma_M(n)}. \quad (\text{E.18})$$

The following recursion can be derived using the above shift invariance properties [7]

$$\Phi_M^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \Phi_{M-1}^{-1}(n-1) \end{bmatrix} + \frac{1}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix} [1 \quad -\mathbf{a}_{M-1}^H(n)], \quad (\text{E.19})$$

where $\mathbf{a}_{M-1}(n)$ is the optimum weight vector for the one-step forward linear predictor of order $M-1$ and can be obtained as

$$\mathbf{a}_{M-1}(n) = \Phi_{M-1}^{-1}(n-1) \mathbf{r}_{M-1}^f(n), \quad (\text{E.20})$$

and $F_{M-1}(n)$ is the corresponding minimum value of the sum of weighted forward a posteriori prediction-error squares defined as

$$F_{M-1}(n) = \sum_{i=1}^n \beta^{n-i} |f_{M-1}(i)|^2, \quad (\text{E.21})$$

with

$$f_{M-1}(i) = u(i) - \mathbf{a}_{M-1}^H(n) \mathbf{u}_{M-1}(i-1). \quad (\text{E.22})$$

Another recursion analog to (E.19) is

$$\Phi_M^{-1}(n) = \begin{bmatrix} \Phi_{M-1}^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} [-\mathbf{b}_{M-1}^H(n) \quad 1], \quad (\text{E.23})$$

where $\mathbf{b}_{M-1}(n)$ is the optimum weight vector for the one-step backward linear predictor of order $M-1$ and can be obtained as

$$\mathbf{b}_{M-1}(n) = \Phi_{M-1}^{-1}(n) \mathbf{r}_{M-1}^b(n), \quad (\text{E.24})$$

and $B_{M-1}(n)$ is the corresponding minimum value of the sum of weighted backward a posteriori prediction-error squares defined as

$$B_{M-1}(n) = \sum_{i=1}^n \beta^{n-i} |b_{M-1}(i)|^2, \quad (\text{E.25})$$

with

$$b_{M-1}(i) = u(i-M+1) - \mathbf{b}_{M-1}^H(n) \mathbf{u}_{M-1}(i). \quad (\text{E.26})$$

Using (E.10), (E.12) and (E.22) in (E.21), the following expression is obtained

$$F_{M-1}(n) = r_{M-1}^{f0}(n) - \mathbf{r}_{M-1}^{fH}(n) \mathbf{a}_{M-1}(n), \quad (\text{E.27})$$

and using (E.9), (E.11) and (E.26) in (E.25),

$$B_{M-1}(n) = r_{M-1}^{b0}(n) - \mathbf{r}_{M-1}^{bH}(n) \mathbf{b}_{M-1}(n). \quad (\text{E.28})$$

Appendix F. Basic Fast Transversal Filter Algorithm

The basic form of the fast algorithm considered in the thesis is given in this appendix. Its derivation is not performed here, since it can be found in many textbooks ([7] or [2] for example). The algorithm presented has been chosen because it exhibits the same basic intermediate variables as those appearing in the matrix-based derivation given in Appendix G. In fact, the most part of the relations and recursions appearing in the fast algorithms are derived in Section G.3. The FTF algorithm that is favored is the *fast a posteriori error sequential technique* (FAEST) of Carayannis et al. [62]. As with the FTF of Cioffi and Kailath [61], the algorithm can be interpreted as a parallel bank of four transversal filter; two for the forward and backward linear predictors $\mathbf{a}_M(n-1)$ and $\mathbf{b}_M(n-1)$, one for the Kalman gain vector $\mathbf{g}_{M+1}(n)$ and one for the actual adaptive weight vector $\hat{\mathbf{w}}_M^\ell(n-1)$. This is illustrated in Figure F.1. Note that it is assumed that a Type II-DRB adaptive system is used. The modifications of the FTF algorithm in order to accommodate a Type I system are straightforward.

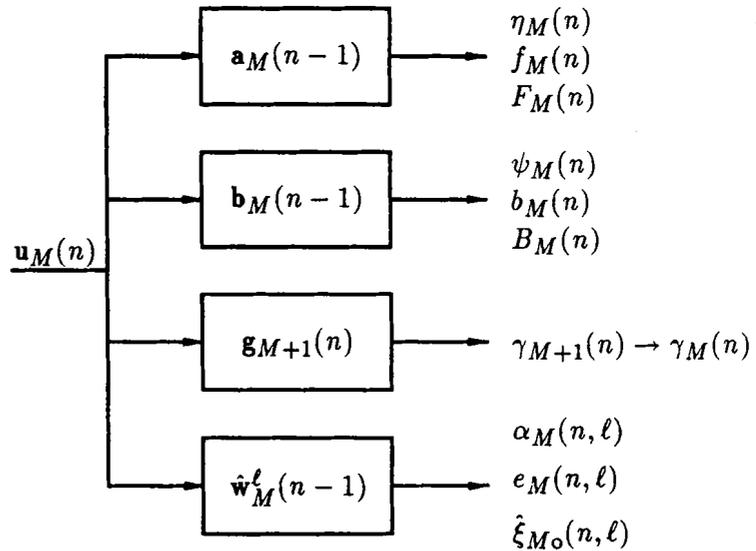


Fig. F.1 Fast Transversal Filter Interpretation

The algorithm is usually separated into two distinct phases; the Kalman gain vector time updating, which is accomplished through the first three transversal filters, and involves the orthogonalization of the input signal with the forward and backward predictors of order M , and the least squares FIR filter time updating, which is performed recursively using the updated Kalman gain vector.

Time updating of the gain vector †

$$\begin{aligned}
 \eta_M(n) &= u(n) - \mathbf{a}_M^H(n-1)\mathbf{u}_M(n-1) \\
 f_M(n) &= \frac{\eta_M(n)}{\gamma_M(n-1)} \\
 \mathbf{a}_M(n) &= \mathbf{a}_M(n-1) + \beta^{-1}\mathbf{g}_M(n-1)f_M^*(n) \\
 F_M(n) &= \beta F_M(n-1) + \eta_M(n)f_M^*(n) \\
 \mathbf{g}_{M+1}(n) &= \begin{bmatrix} 0 \\ \mathbf{g}_M(n-1) \end{bmatrix} + \frac{\eta_M(n)}{F_M(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}_M(n-1) \end{bmatrix} \\
 \mathbf{g}_M(n) &= [\mathbf{g}_{M+1}(n)]_M + g_{M+1,M+1}(n)\mathbf{b}_M(n-1) \\
 \gamma_{M+1}(n) &= \gamma_M(n-1) + \frac{|\eta_M(n)|^2}{\beta F_M(n-1)} \\
 \psi_M(n) &= g_{M+1,M+1}(n)B_M(n-1) \\
 \gamma_M(n) &= \gamma_{M+1}(n) - \beta^{-1}g_{M+1,M+1}(n)\psi_M^*(n) \\
 b_M(n) &= \frac{\psi_M(n)}{\gamma_M(n)} \\
 \mathbf{b}_M(n) &= \mathbf{b}_M(n-1) + \beta^{-1}\mathbf{g}_M(n)b_M^*(n) \\
 B_M(n) &= \beta B_M(n-1) + \psi_M(n)b_M^*(n)
 \end{aligned} \tag{F.1}$$

Time updating of the LS FIR filter

$$\begin{aligned}
 \alpha_M(n, \ell) &= r(n+\ell) - \hat{\mathbf{w}}_M^\ell(n-1)\mathbf{u}_M(n) \\
 e_M(n, \ell) &= \frac{\alpha_M(n, \ell)}{\gamma_M(n)} \\
 \hat{\mathbf{w}}_M^\ell(n) &= \hat{\mathbf{w}}_M^\ell(n-1) + \beta^{-1}\mathbf{g}_M(n)e_M^*(n, \ell)
 \end{aligned} \tag{F.2}$$

† The notation $[\mathbf{v}]_m$ stands for the vector made of the m first components of the vector \mathbf{v} and $[\mathbf{v}]_m$ for the vector made of the m last components of the vector \mathbf{v} .

Appendix G. Matrix-based Derivation of the Error and Weight Vector Recursions: Type II-DRB

It is assumed that the least squares weight vector $\hat{\mathbf{w}}_M^\ell(n)$ and the corresponding least error squares $\hat{\xi}_{M_o}(n, \ell)$ are available at iteration n . It is desired to compute, from these values, the least error squares for $\ell - 1$ and $\ell + 1$, and the least squares weight vector corresponding to the lowest error. Recursions for the error are first developed, followed by similar recursions for $\hat{\mathbf{w}}_M^{\ell-1}(n)$ and $\hat{\mathbf{w}}_M^{\ell+1}(n)$. The derivations follow closely the ones presented in [63] for a fixed-length block of data.

G.1 Recursions for the Error

The least squares error, for lag ℓ in the reference path, can be expressed as [7]

$$\hat{\xi}_{M_o}(n, \ell) = \hat{\xi}_d(n, \ell) - \boldsymbol{\theta}_M^{\ell H}(n) \hat{\mathbf{w}}_M^\ell(n), \quad (\text{G.1})$$

where

$$\begin{aligned} \hat{\xi}_d(n, \ell) &= \sum_{i=1}^n \beta^{n-i} |r(i + \ell)|^2 \\ &= \beta \hat{\xi}_d(n-1, \ell) + |r(n + \ell)|^2. \end{aligned} \quad (\text{G.2})$$

Use of (E.4) and (E.19) in (5.20) gives

$$\begin{aligned} \hat{\mathbf{w}}_M^\ell(n) &= \\ & \left\{ \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \Phi_{M-1}^{-1}(n-1) \end{bmatrix} + \frac{1}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix} \begin{bmatrix} 1 & -\mathbf{a}_{M-1}^H(n) \end{bmatrix} \right\} \begin{bmatrix} \tilde{\boldsymbol{\theta}}_M^\ell(n) \\ \boldsymbol{\theta}_{M-1}^{\ell+1}(n-1) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) \end{bmatrix} + \frac{1}{F_{M-1}(n)} \begin{bmatrix} \tilde{\boldsymbol{\theta}}_M^\ell(n) - \mathbf{a}_{M-1}^H(n) \boldsymbol{\theta}_{M-1}^{\ell+1}(n-1) \\ -\mathbf{a}_{M-1}(n) (\tilde{\boldsymbol{\theta}}_M^\ell(n) - \mathbf{a}_{M-1}^H(n) \boldsymbol{\theta}_{M-1}^{\ell+1}(n-1)) \end{bmatrix}. \end{aligned} \quad (\text{G.3})$$

Noting that

$$w_{1M}^\ell(n) = \frac{\tilde{\theta}_M^\ell(n) - \mathbf{a}_{M-1}^H(n) \boldsymbol{\theta}_{M-1}^{\ell+1}(n-1)}{F_{M-1}(n)} \quad (\text{G.4})$$

is the first component of $\hat{\mathbf{w}}_M^\ell(n)$, (G.3) can be written as

$$\hat{\mathbf{w}}_M^\ell(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) \end{bmatrix} + \begin{bmatrix} \hat{w}_{1M}^\ell(n) \\ -\mathbf{a}_{M-1}(n) w_{1M}^\ell(n) \end{bmatrix}. \quad (\text{G.5})$$

Use of (E.4) and (G.5) in (G.1) gives

$$\hat{\xi}_{M_o}(n, \ell) = \hat{\xi}_d(n, \ell)$$

$$\begin{aligned}
& -[\tilde{\theta}_M^{\ell*}(n) \quad \theta_{M-1}^{(\ell+1)H}(n-1)] \left\{ \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) \end{bmatrix} + \begin{bmatrix} \hat{w}_{1M}^\ell(n) \\ -\mathbf{a}_{M-1}(n)w_{1M}^\ell(n) \end{bmatrix} \right\} \\
& = \hat{\xi}_d(n, \ell) - \hat{w}_{1M}^\ell(n)[\tilde{\theta}_M^{\ell*}(n) - \theta_{M-1}^{(\ell+1)H}(n-1)\mathbf{a}_{M-1}(n)] - \theta_{M-1}^{(\ell+1)H}(n-1)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1). \quad (\text{G.6})
\end{aligned}$$

Write $\hat{\xi}_d(n, \ell)$ as

$$\begin{aligned}
\hat{\xi}_d(n, \ell) &= \sum_{i=0}^{n-1} \beta^{n-1-i} |r(i+\ell+1)|^2 \\
&= \sum_{i=1}^{n-1} \beta^{n-1-i} |r(i+\ell+1)|^2 + \beta^{n-1} |r(\ell+1)|^2 \\
&= \hat{\xi}_d(n-1, \ell+1) + \beta^{n-1} |r(\ell+1)|^2. \quad (\text{G.7})
\end{aligned}$$

Use of (G.4) and (G.7) in (G.6) gives

$$\begin{aligned}
\hat{\xi}_{M_o}(n, \ell) &= \hat{\xi}_d(n-1, \ell+1) \\
&\quad - \theta_{M-1}^{(\ell+1)H}(n-1)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - F_{M-1}(n)|\hat{w}_{1M}^\ell(n)|^2 + \beta^{n-1}|r(\ell+1)|^2 \\
&= \hat{\xi}_{(M-1)_o}(n-1, \ell+1) - F_{M-1}(n)|\hat{w}_{1M}^\ell(n)|^2 + \beta^{n-1}|r(\ell+1)|^2. \quad (\text{G.8})
\end{aligned}$$

Therefore, from (G.8), a first recursion on the least error squares is

$$\hat{\xi}_{(M-1)_o}(n-1, \ell+1) = \hat{\xi}_{M_o}(n, \ell) + F_{M-1}(n)|\hat{w}_{1M}^\ell(n)|^2 \quad \text{for} \quad \beta^{n-1} \approx 0. \quad (\text{G.9})$$

In order to obtain a relation involving $\hat{\xi}_{(M-1)_o}(n-1, \ell+1)$, extend (G.1) to $\ell+1$ and $M-1$

$$\hat{\xi}_{(M-1)_o}(n, \ell+1) = \hat{\xi}_d(n, \ell+1) - \theta_{M-1}^{(\ell+1)H}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n). \quad (\text{G.10})$$

Using (E.14) and (E.18) to express $\hat{\mathbf{w}}_{M-1}^{\ell+1}(n)$

$$\begin{aligned}
\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) &= \Phi_{M-1}^{-1}(n)\theta_{M-1}^{\ell+1}(n) \\
&= [\beta^{-1}\Phi_{M-1}^{-1}(n-1) - \beta^{-2}\frac{\mathbf{g}_{M-1}(n)\mathbf{g}_{M-1}^H(n)}{\gamma_{M-1}(n)}][\beta\theta_{M-1}^{\ell+1}(n-1) + \mathbf{u}_{M-1}(n)r^*(n+\ell+1)] \\
&= \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \frac{\mathbf{g}_{M-1}(n)\mathbf{g}_{M-1}^H(n)\theta_{M-1}^{\ell+1}(n-1)}{\beta\gamma_{M-1}(n)} \\
&+ \beta^{-1}\Phi_{M-1}^{-1}(n-1)\mathbf{u}_{M-1}(n)r^*(n+\ell+1) - \frac{\beta^{-2}\mathbf{g}_{M-1}(n)\mathbf{g}_{M-1}^H(n)\mathbf{u}_{M-1}(n)r^*(n+\ell+1)}{\gamma_{M-1}(n)}. \quad (\text{G.11})
\end{aligned}$$

Using (E.16) and (E.17) and after some manipulations, (G.11) simplifies to

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) = \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \frac{\mathbf{g}_{M-1}(n)}{\beta\gamma_{M-1}(n)}[\mathbf{u}_{M-1}^H(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - r^*(n+\ell+1)]. \quad (\text{G.12})$$

Using (E.14) and (G.12) in (G.10) and noting that

$$\hat{\xi}_d(n, \ell + 1) = \beta \hat{\xi}_d(n - 1, \ell + 1) + |r(n + \ell + 1)|^2 \quad (\text{G.13})$$

gives, after some manipulations,

$$\begin{aligned} \hat{\xi}_{(M-1)_o}(n, \ell + 1) &= \beta \hat{\xi}_{(M-1)_o1}(n - 1, \ell + 1) \\ &\quad + \boldsymbol{\theta}_{M-1}^{(\ell+1)H} \mathbf{g}_{M-1}(n) \frac{v}{\gamma_{M-1}(n)} \\ &\quad + r(n + \ell + 1) \frac{v}{\gamma_{M-1}(n)} [\gamma_{M-1}^*(n) - \gamma_{M-1}(n) - 1], \end{aligned} \quad (\text{G.14})$$

where v is defined as

$$v = \mathbf{u}_{M-1}^H(n) \hat{\mathbf{w}}_{M-1}^{\ell+1}(n - 1) - r^*(n + \ell + 1). \quad (\text{G.15})$$

Note that $\Phi_M(n)$ is Hermitian symmetric, i.e.

$$\Phi_M(n) = \Phi_M^H(n), \quad (\text{G.16})$$

which implies that $\Phi_M^{-1}(n)$ is also Hermitian symmetric [7]. This, in turn, implies that $\Phi_M^{-1}(n)$ is positive semi-definite with *real* eigenvalues. Then

$$\gamma_M(n) = 1 + \beta^{-1} \mathbf{u}_M^H(n) \Phi_M^{-1}(n - 1) \mathbf{u}_M(n) \quad (\text{G.17})$$

is real if β is real. Therefore, (G.14) simplifies to

$$\hat{\xi}_{(M-1)_o}(n, \ell + 1) = \beta \hat{\xi}_{(M-1)_o}(n - 1, \ell + 1) + [\boldsymbol{\theta}_{M-1}^{(\ell+1)H} \mathbf{g}_{M-1}(n) - r(n + \ell + 1)] \frac{v}{\gamma_{M-1}(n)}. \quad (\text{G.18})$$

Using (E.18) and (5.20) in (G.18) gives

$$\hat{\xi}_{(M-1)_o}(n, \ell + 1) = \beta \hat{\xi}_{(M-1)_o}(n - 1, \ell + 1) + v^* \frac{v}{\gamma_{M-1}(n)}, \quad (\text{G.19})$$

i.e.

$$\hat{\xi}_{(M-1)_o}(n, \ell + 1) = \beta \hat{\xi}_{(M-1)_o}(n - 1, \ell + 1) + \frac{|\mathbf{u}_{M-1}^H(n) \hat{\mathbf{w}}_{M-1}^{\ell+1}(n - 1) - r^*(n + \ell + 1)|^2}{\gamma_{M-1}(n)}, \quad (\text{G.20})$$

which is the second recursion required. It allows the computation of $\hat{\xi}_{(M-1)_o}(n, \ell + 1)$ from $\hat{\xi}_{(M-1)_o}(n - 1, \ell + 1)$, while (G.10) allows the computation of $\hat{\xi}_{(M-1)_o}(n - 1, \ell + 1)$ from $\hat{\xi}_{M_o}(n, \ell)$. All is required is a relation linking $\hat{\xi}_{(M-1)_o}(n, \ell + 1)$ to $\hat{\xi}_{M_o}(n, \ell + 1)$.

This relation can be obtained by first computing a relation similar to (G.5) with the help of (E.2) and (E.23) in

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \Phi_M^{-1}(n) \boldsymbol{\theta}_M^{\ell+1}(n). \quad (\text{G.21})$$

This gives

$$\begin{aligned}\hat{\mathbf{w}}_M^{\ell+1}(n) &= \left\{ \begin{bmatrix} \Phi_{M-1}^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{b}_{M-1}^H(n) & 1 \end{bmatrix} \right\} \begin{bmatrix} \boldsymbol{\theta}_{M-1}^{\ell+1}(n) \\ \theta_M^{\ell+1}(n) \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{1}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} [-\mathbf{b}_{M-1}^H(n)\boldsymbol{\theta}_{M-1}^{\ell+1}(n) + \theta_M^{\ell+1}(n)]. \quad (\text{G.22})\end{aligned}$$

Noting that (using (E.24))

$$\begin{aligned}-\mathbf{b}_{M-1}^H(n)\boldsymbol{\theta}_{M-1}^{\ell+1}(n) &= -\mathbf{r}_{M-1}^{bH}(n)\Phi_{M-1}^{-1}(n-1)\boldsymbol{\theta}_{M-1}^{\ell+1}(n) \\ &= -\mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n),\end{aligned} \quad (\text{G.23})$$

equation (G.22) can be written as

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{1}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} [-\mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) + \theta_M^{\ell+1}(n)]. \quad (\text{G.24})$$

Now, use (E.2) and (G.24) in (G.1) for $\ell + 1$

$$\begin{aligned}\hat{\xi}_{M_o}(n, \ell + 1) &= \hat{\xi}_d(n, \ell + 1) - \left[\boldsymbol{\theta}_{M-1}^{(\ell+1)H}(n) \quad \theta_M^{(\ell+1)*}(n) \right] \\ &\quad \left\{ \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{1}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix} [-\mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) + \theta_M^{\ell+1}(n)] \right\} \\ &= \hat{\xi}_d(n, \ell + 1) - \boldsymbol{\theta}_{M-1}^{(\ell+1)H}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) - \frac{1}{B_{M-1}(n)} |\theta_M^{\ell+1}(n) - \mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n)|^2. \quad (\text{G.25})\end{aligned}$$

Using (G.1) in (G.25) gives

$$\hat{\xi}_{M_o}(n, \ell + 1) = \hat{\xi}_{(M-1)_o}(n, \ell + 1) - \frac{1}{B_{M-1}(n)} |\theta_M^{\ell+1}(n) - \mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n)|^2. \quad (\text{G.26})$$

This expression can be written in a different form by noting that

$$\begin{aligned}\boldsymbol{\theta}_M^{\ell+1}(n) - \mathbf{r}_{M-1}^{bH}(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) &= [-\mathbf{b}_{M-1}^H(n) \quad 1]\boldsymbol{\theta}_M^{\ell+1}(n) \\ &= [-\mathbf{b}_{M-1}^H(n) \quad 1] \sum_{i=1}^n \beta^{n-i} \mathbf{u}_M(i) r^*(i + \ell + 1) \\ &= \sum_{i=1}^n \beta^{n-i} [u(i - M + 1) - \mathbf{b}_{M-1}^H(n)\mathbf{u}_{M-1}(i)] r^*(i + \ell + 1) \\ &= \sum_{i=1}^n \beta^{n-i} b_{M-1}(i) r^*(i + \ell + 1).\end{aligned} \quad (\text{G.27})$$

Define

$$v_{M-1}^{b(\ell+1)}(n) = \sum_{i=1}^n \beta^{n-i} b_{M-1}(i) r^*(i + \ell + 1). \quad (\text{G.28})$$

Then (G.26) is written in the form

$$\hat{\xi}_{M_o}(n, \ell + 1) = \hat{\xi}_{(M-1)_o}(n, \ell + 1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}(n)}. \quad (\text{G.29})$$

This last expression is the third necessary error recursion. Collecting (G.9), (G.20) and (G.29), the recursions for computing $\hat{\xi}_{M_o}(n, \ell + 1)$ from $\hat{\xi}_{M_o}(n, \ell)$ are

$$\begin{aligned} \hat{\xi}_{(M-1)_o}(n-1, \ell+1) &= \hat{\xi}_{M_o}(n, \ell) + F_{M-1}(n) |\hat{w}_{1M}^\ell(n)|^2 \\ \hat{\xi}_{(M-1)_o}(n, \ell+1) &= \beta \hat{\xi}_{(M-1)_o}(n-1, \ell+1) + \frac{|\mathbf{u}_{M-1}^H(n) \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - r^*(n+\ell+1)|^2}{\gamma_{M-1}(n)} \\ \hat{\xi}_{M_o}(n, \ell+1) &= \hat{\xi}_{(M-1)_o}(n, \ell+1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}(n)}. \end{aligned} \quad (\text{G.30})$$

Using the above expressions in reverse order gives the backward computation of the error;

$$\begin{aligned} \hat{\xi}_{(M-1)_o}(n, \ell) &= \hat{\xi}_{M_o}(n, \ell) + \frac{|v_{M-1}^{b\ell}(n)|^2}{B_{M-1}(n)} \\ \hat{\xi}_{(M-1)_o}(n-1, \ell) &= \beta^{-1} \hat{\xi}_{(M-1)_o}(n, \ell) - \beta^{-1} \frac{|\mathbf{u}_{M-1}^H(n) \hat{\mathbf{w}}_{M-1}^\ell(n-1) - r^*(n+\ell)|^2}{\gamma_{M-1}(n)} \\ \hat{\xi}_{M_o}(n, \ell-1) &= \hat{\xi}_{(M-1)_o}(n-1, \ell) - F_{M-1}(n) |\hat{w}_{1M}^{\ell-1}(n)|^2. \end{aligned} \quad (\text{G.31})$$

G.2 Recursions for the LS Weight Vector

The recursions for the upward weight vector computation were all derived in the previous section on error recursions. The recursions for downward computations are obtained by applying the upward recursions in reverse order.

G.2.1 Recursions for the upward weight vector computation

The first recursion on the weight vector is obtained from (G.5) and can be written as

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}(n) \hat{w}_{1M}^\ell(n), \quad (\text{G.32})$$

where $[\hat{\mathbf{w}}_M^\ell(n)]_{M-1}$ is defined as the $(M-1)$ -vector corresponding to the last $M-1$ components of $\hat{\mathbf{w}}_M^\ell(n)$.

The second and third recursions are given by (G.12) and (G.24) respectively. The set of recursions is therefore

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{a}_{M-1}(n)\hat{w}_{1M}^\ell(n) \quad (\text{G.33})$$

$$\hat{\mathbf{w}}_{M-1}^{\ell+1}(n) = \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \frac{\mathbf{g}_{M-1}(n)}{\beta\gamma_{M-1}(n)} [\mathbf{u}_{M-1}^H(n)\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - r^*(n+\ell+1)] \quad (\text{G.34})$$

$$\hat{\mathbf{w}}_M^{\ell+1}(n) = \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix}. \quad (\text{G.35})$$

G.2.2 Recursions for the downward weight vector computation

Use the upward recursions in reverse order. The corresponding set of recursions is

$$\hat{\mathbf{w}}_{M-1}^\ell(n) = [\hat{\mathbf{w}}_M^\ell(n)]_{M-1} + \mathbf{b}_{M-1}(n)\hat{w}_{MM}^\ell(n) \quad (\text{G.36})$$

$$\hat{\mathbf{w}}_M^{\ell-1}(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^\ell(n-1) \end{bmatrix} + \frac{v_{M-1}^{f(\ell-1)}(n)}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix}, \quad (\text{G.37})$$

where $[\hat{\mathbf{w}}_M^\ell(n)]_{M-1}$ is defined as the $(M-1)$ -vector corresponding to the *first* components of $\hat{\mathbf{w}}_M^\ell(n)$, $\hat{w}_{MM}^\ell(n)$ is the M^{th} component of the same vector and $v_{M-1}^{f(\ell-1)}(n)$ is defined as

$$\begin{aligned} v_{M-1}^{f(\ell-1)}(n) &= \tilde{\theta}_M^{\ell-1}(n) - \mathbf{r}_{M-1}^{fH}(n)\hat{\mathbf{w}}_{M-1}^\ell(n-1) \\ &= \tilde{\theta}_M^{\ell-1}(n) - \mathbf{a}_{M-1}^H(n)\theta_{M-1}^\ell(n-1) \\ &= \sum_{i=1}^n \beta^{n-i} f_{M-1}(i)r^*(i+\ell-1). \end{aligned} \quad (\text{G.38})$$

G.3 Auxiliary recursions

Some auxiliary recursions necessary in the error and vector recursions are developed in this section.

o Recursion for $\mathbf{g}_{M-1}(n)$

Use (E.19) and (E.1) in (E.16)

$$\mathbf{g}_M(n) = \left\{ \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \Phi_{M-1}^{-1}(n-2) \end{bmatrix} + \frac{1}{F_{M-1}(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n-1) \end{bmatrix} \begin{bmatrix} 1 & -\mathbf{a}_{M-1}^H(n-1) \end{bmatrix} \right\} \begin{bmatrix} u(n) \\ \mathbf{u}_{M-1}(n-1) \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \mathbf{g}_{M-1}(n-1) \end{bmatrix} + \frac{1}{F_{M-1}(n-1)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n-1) \end{bmatrix} [u(n) - \mathbf{a}_{M-1}^H(n-1)\mathbf{u}_{M-1}(n-1)]. \quad (\text{G.39})$$

Now use (E.1) and (E.23) in (E.16)

$$\begin{aligned} \mathbf{g}_M(n) &= \left\{ \begin{bmatrix} \Phi_{M-1}^{-1}(n-1) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_{M-1}(n-1)} \begin{bmatrix} -\mathbf{b}_{M-1}(n-1) \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{b}_{M-1}^H(n-1) & 1 \end{bmatrix} \right\} \\ &\quad \begin{bmatrix} \mathbf{u}_{M-1}(n) \\ u(n-M+1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{g}_{M-1}(n) \\ 0 \end{bmatrix} + \frac{1}{B_{M-1}(n-1)} \begin{bmatrix} -\mathbf{b}_{M-1}(n-1) \\ 1 \end{bmatrix} [-\mathbf{b}_{M-1}^H(n-1)\mathbf{u}_{M-1}(n) + u(n-M+1)], \end{aligned} \quad (\text{G.40})$$

i.e.

$$\mathbf{g}_{M-1}(n) = [\mathbf{g}_M(n)]_{M-1} + g_{MM}(n)\mathbf{b}_{M-1}(n-1). \quad (\text{G.41})$$

Equations (G.39) and (G.41) are the recursions for $\mathbf{g}_{M-1}(n)$.

◦ **Recursions for $F_{M-1}(n)$ and $B_{M-1}(n)$**

The recursions for $F_{M-1}(n)$ and $B_{M-1}(n)$ are [7]

$$F_{M-1}(n) = \beta F_{M-1}(n-1) + \eta_{M-1}(n)f_{M-1}^*(n) \quad (\text{G.42})$$

and

$$B_{M-1}(n) = \beta B_{M-1}(n-1) + \psi_{M-1}(n)b_{M-1}^*(n), \quad (\text{G.43})$$

where $\eta_{M-1}(n)$ and $\psi_{M-1}(n)$ are respectively the forward *a priori* prediction error and the backward *a priori* prediction error defined as

$$\begin{aligned} \eta_{M-1}(n) &= [1 \quad -\mathbf{a}_{M-1}^H(n-1)]\mathbf{u}_M(n) \\ &= u(n) - \mathbf{a}_{M-1}^H(n-1)\mathbf{u}_{M-1}(n-1) \end{aligned} \quad (\text{G.44})$$

$$\begin{aligned} \psi_{M-1}(n) &= [-\mathbf{b}_{M-1}^H(n-1) \quad 1]\mathbf{u}_M(n) \\ &= u(n-M+1) - \mathbf{b}_{M-1}^H(n-1)\mathbf{u}_{M-1}(n). \end{aligned} \quad (\text{G.45})$$

◦ **Recursions for the forward and backward predictors**

For the forward case, use (E.18) and (E.10) in (E.20)

$$\begin{aligned} \mathbf{a}_{M-1}(n) &= \left[\beta^{-1}\Phi_{M-1}^{-1}(n-2) - \beta^{-2} \frac{\mathbf{g}_{M-1}(n-1)\mathbf{g}_{M-1}^H(n-1)}{\gamma_{M-1}(n-1)} \right] \\ &\quad \left[\beta \mathbf{r}_{M-1}^f(n-1) + \mathbf{u}_{M-1}(n-1)u^*(n) \right]. \end{aligned}$$

Using (E.16), (E.17) and (E.20) and after some manipulations, the recursion for the forward predictor is

$$\mathbf{a}_{M-1}(n) = \mathbf{a}_{M-1}(n-1) + \frac{\mathbf{g}_{M-1}(n-1)}{\beta\gamma_{M-1}(n-1)} [u^*(n) - \mathbf{u}_{M-1}^H(n-1)\mathbf{a}_{M-1}(n-1)], \quad (\text{G.46})$$

i.e.

$$\mathbf{a}_{M-1}(n) = \mathbf{a}_{M-1}(n-1) + \frac{\mathbf{g}_{M-1}(n-1)}{\beta\gamma_{M-1}(n-1)} \eta_{M-1}^*(n). \quad (\text{G.47})$$

The recursion for the backward error is obtained in a similar way and is

$$\mathbf{b}_{M-1}(n) = \mathbf{b}_{M-1}(n-1) + \frac{\mathbf{g}_{M-1}(n)}{\beta\gamma_{M-1}(n)} \psi_{M-1}^*(n). \quad (\text{G.48})$$

o **Recursions for $\gamma_M(n)$ and $\gamma_{M-1}(n)$**

In order to establish recursions for $\gamma_M(n)$ and $\gamma_{M-1}(n)$, the following identities are necessary [7]

$$f_{M-1}(n) = \frac{\eta_{M-1}(n)}{\gamma_{M-1}(n-1)} \quad (\text{G.49})$$

$$b_{M-1}(n) = \frac{\psi_{M-1}(n)}{\gamma_{M-1}(n)} \quad (\text{G.50})$$

$$g_{MM}(n) = \frac{\psi_{M-1}(n)}{B_{M-1}(n-1)}. \quad (\text{G.51})$$

Then, using (E.17) and (G.39)

$$\begin{aligned} \gamma_M(n) &= 1 + \beta^{-1} \mathbf{g}_{M-1}^H(n) \mathbf{u}_{M-1}(n) \\ &= \gamma_{M-1}(n-1) + \frac{|\eta_{M-1}(n)|^2}{\beta F_{M-1}(n-1)}. \end{aligned} \quad (\text{G.52})$$

Also,

$$\begin{aligned} \gamma_M(n) &= \gamma_{M-1}(n) + \frac{|\psi_{M-1}(n)|^2}{\beta B_{M-1}(n)} \\ &= \gamma_{M-1}(n) + \beta^{-1} g_{MM}(n) \psi_{M-1}^*(n). \end{aligned} \quad (\text{G.53})$$

o **Recursions for $v_m^{f\ell}(n)$ and $v_m^{b\ell}(n)$**

From (G.38), $v_m^{f\ell}(n)$ is defined as

$$v_m^{f\ell}(n) = [1 \quad - \mathbf{a}_m^H(n)] \boldsymbol{\theta}_{m+1}^\ell(n). \quad (\text{G.54})$$

From (G.47) and (G.49), one obtains

$$\mathbf{a}_m(n) = \mathbf{a}_m(n-1) + \frac{\mathbf{g}_m(n-1)}{\beta} f_m^*(n). \quad (\text{G.55})$$

Then, using (E.14) and (G.55) in (G.54),

$$\begin{aligned} v_m^{f\ell}(n) &= \beta [1 \quad - \mathbf{a}_m^H(n-1)] \boldsymbol{\theta}_{m+1}^\ell(n-1) + [1 \quad - \mathbf{a}_m^H(n-1)] \mathbf{u}_{m+1}(n) r^*(n+\ell) \\ &\quad - [0 \quad \mathbf{g}_m^H(n-1)] \boldsymbol{\theta}_{m+1}^\ell(n) \beta^{-1} f_m(n). \end{aligned}$$

Using (E.1) and (E.4),

$$\begin{aligned}
v_m^{f\ell}(n) &= \beta v_m^{f\ell}(n-1) + [u(n) - \mathbf{a}_m^H(n-1)\mathbf{u}_m(n-1)]r^*(n+\ell) \\
&\quad - \beta^{-1}\mathbf{g}_m^H(n-1)\boldsymbol{\theta}_m^{\ell+1}(n-1)f_m(n) \\
&= \beta v_m^{f\ell}(n-1) + \eta_m(n)r^*(n+\ell) \\
&\quad - \beta^{-1}\mathbf{g}_m^H(n-1)\boldsymbol{\theta}_m^{\ell+1}(n-1)f_m(n).
\end{aligned} \tag{G.56}$$

Using the definition of $\mathbf{g}_m(n)$, (see equation (E.16)), (G.56) can be written as

$$v_m^{f\ell}(n) = \beta v_m^{f\ell}(n-1) + \eta_m(n)[r^*(n+\ell) - \frac{\mathbf{u}_m(n-1)\Phi_m^{-1}(n-2)\boldsymbol{\theta}_m^{\ell+1}(n-1)}{\beta\gamma_m(n-1)}]. \tag{G.57}$$

But using (E.14), the second term in brackets is equal to $r^*(n+\ell) - e_m^*(n-1, \ell+1)$ and (G.57) becomes

$$v_m^{f\ell}(n) = \beta v_m^{f\ell}(n-1) + \eta_m(n)e_m^*(n-1, \ell+1). \tag{G.58}$$

Similarly, the recursion for $v_m^{b\ell}(n)$ is found to be

$$\begin{aligned}
v_m^{b\ell}(n) &= [-\mathbf{b}_m^H(n) \quad 1]\boldsymbol{\theta}_{m+1}^\ell(n) \\
&= \beta v_m^{b\ell}(n-1) + \psi_m(n)e_m^*(n, \ell).
\end{aligned} \tag{G.59}$$

o **Recursion for $\hat{\mathbf{w}}_M^\ell(n)$**

A recursion on $\hat{\mathbf{w}}_M^\ell(n)$ is obtained by starting from (5.20) and proceeding as in the derivation of (G.12). It is

$$\hat{\mathbf{w}}_M^\ell(n) = \hat{\mathbf{w}}_M^\ell(n-1) + \frac{\mathbf{g}_M(n)}{\beta\gamma_M(n)}[r^*(n+\ell) - \mathbf{u}_M^H(n)\hat{\mathbf{w}}_M^\ell(n-1)]. \tag{G.60}$$

Now, the *a priori* estimation error is

$$\alpha_M(n, \ell) = r(n+\ell) - \hat{\mathbf{w}}_M^{\ell H}(n-1)\mathbf{u}_M(n), \tag{G.61}$$

and the *a posteriori* estimation error is

$$e_M(n, \ell) = r(n+\ell) - \hat{\mathbf{w}}_M^{\ell H}(n)\mathbf{u}_M(n). \tag{G.62}$$

Then

$$\hat{\mathbf{w}}_M^\ell(n) = \hat{\mathbf{w}}_M^\ell(n-1) + \beta^{-1}\mathbf{g}_M(n)\frac{e_M^*(n, \ell)}{\gamma_M(n)}. \tag{G.63}$$

It can be shown that [7]

$$e_M(n, \ell) = \frac{\alpha_M(n, \ell)}{\gamma_M(n)}, \tag{G.64}$$

and therefore

$$\hat{\mathbf{w}}_M^\ell(n) = \hat{\mathbf{w}}_M^\ell(n-1) + \beta^{-1}\mathbf{g}_M(n)e_M^*(n, \ell). \tag{G.65}$$

o **Recursion for $\hat{\xi}_{M_o}(n, \ell)$**

The recursion for the minimum error is [7]

$$\hat{\xi}_{M_o}(n, \ell) = \beta\hat{\xi}_{M_o}(n-1, \ell) + \alpha_M^*(n, \ell)e_M(n, \ell).$$

References

1. J. G. Proakis, *Digital Communications*, McGraw-Hill, 1983.
2. M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms and Applications*, Kluwer Academic Publishers, 1984.
3. B. Widrow et al., "Adaptive noise cancelling: principles and applications," *Proc. IEEE*, vol. 63, pp. 1692–1716, December 1975.
4. L. Ljung, *System Identification: Theory for the user*, Prentice-Hall, 1987.
5. S. L. Marple, jr, *Digital Spectral Analysis with Applications*, Prentice-Hall, 1987.
6. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.
7. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 1986.
8. E. A. Robinson and S. Treitel, *Geophysical Signal Analysis*, Prentice-Hall, 1980.
9. V. K. Madisetti, D. G. Messerschmitt and N. Nordstrom, "Dynamically-reduced complexity implementation of echo cancelers," *IEEE 1986 International Conference on Acoustic, Speech and Signal Processing*, pp. 1313–1316, 1986.
10. B. Widrow, J. M. McCool, M. G. Larimore and C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151–1162, August 1976.
11. L. Barbeau et al., "Speech enhancement in the presence of interfering music and noise," *Rapport technique de l'INRS-Télécommunications*, no. 87-09, January 1987.
12. D. O'Shaughnessy, P. Kabal, D. Bernardi, L. Barbeau, C.-C. Chu and J.-L. Moncet, "Applying speech enhancement to audio surveillance," *Journal of Forensic Sciences*, to appear in the September 1990 issue.
13. S. U. H. Qureshi, "Timing recovery for equalized partial-response systems," *IEEE Trans. on Communications*, pp. 1326–1331, December 1976.
14. S. U. H. Qureshi, "Adaptive equalization," *Proc. of the IEEE*, vol. 73, pp. 1349–1387, September 1985.
15. C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 320–327, August 1976.
16. J. A. Stuller, "Maximum-likelihood estimation of time-varying delay—part I," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 300–313, March 1987.
17. B. Champagne, M. Eizenman and S. Pasupathy, "Exact maximum likelihood time delay estimation," *IEEE 1990 International Conference on Acoustic, Speech and Signal Processing*, pp. 2633–2636, 1989.
18. H. Meyr and G. Spies, "The structure and performance of estimators for real-time estimation of randomly varying time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 81–94, February 1984.
19. H. Messer, "A unified approach to closed-loop time-delay estimation systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 854–861, June 1988.
20. A. V. Oppenheim, A. S. Willsky and I. T. Young, *Signals and Systems*, Prentice-Hall, 1983.

21. F. A. Reed, P. L. Feintuch and N. J. Bershad, "Time delay estimation using the LMS adaptive filter-static behavior," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 561-570, June 1981.
22. J. Krolik, M. Eizenman and S. Pasupathy, "Time delay estimation of signals with uncertain spectra," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1801-1811, December 1988.
23. P. L. Feintuch, N. J. Bershad and F. A. Reed, "Time delay estimation using the LMS adaptive filter-dynamic behavior," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 571-576, June 1981.
24. D. H. Youn, N. Ahmed and G. C. Carter, "On using the LMS algorithm for time delay estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 798-801, October 1982.
25. Y. T. Chan, J. M. Riley and J. B. Plant, "A parameter estimation approach to time-delay estimation and signal detection," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 8-16, February 1980.
26. Y. T. Chan, J. M. Riley and J. B. Plant, "Modeling of time delay and its application to estimation of nonstationary delays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 577-581, June 1981.
27. D. H. Youn, N. Ahmed and G. C. Carter, "An adaptive approach for time delay estimation of band-limited signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 780-784, June 1983.
28. D. M. Etter and S. D. Stearn, "Adaptive estimation of time delays in sampled data systems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 582-587, June 1981.
29. H. Messer and Y. Bar-Ness, "Closed-loop least mean square time-delay estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 413-424, April 1987.
30. J. O. Smith, B. Friedlander, "Adaptive interpolated time-delay estimation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-21, pp. 180-199, March 1985.
31. D. Luenberger, *Linear and Nonlinear Programming, 2nd edition*, Addison-Wesley, 1984.
32. C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Prentice-Hall, 1985.
33. E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1097-1110, October 1986.
34. 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," *IEEE 1990 International Symposium on Circuits and Systems*, pp. 266-270, 1990.
35. T. A. Ramstad, "Digital methods for conversion between arbitrary sampling frequencies," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 577-591, June 1984.
36. H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems," *IEEE Trans. on Communication Technology*, vol. COM-19, pp. 268-280, June 1971.
37. R. W. Chang, "Joint optimization of automatic equalization and carrier acquisition for digital communication," *The Bell System Technical Journal*, pp. 1069-1105, July-August 1970.

38. D. D. Falconer, "Jointly adaptive equalization and carrier recovery in two-dimensional digital communication systems," *The Bell System Technical Journal*, vol. 55, pp. 317-334, March 1976.
39. D. D. Falconer, "Analysis of a gradient algorithm for simultaneous equalization and carrier phase recovery," *The Bell System Technical Journal*, vol. 55, pp. 409-428, April 1976.
40. S. U. H. Qureshi, "Adjustment of the position of the reference tap of an adaptive equalizer," *IEEE Trans. on Communications*, pp. 1046-1052, September 1973.
41. D. Boudreau and P. Kabal, "Joint gradient-based time delay estimation and adaptive filtering," *IEEE 1990 International Symposium on Circuits and Systems*, pp. 3165-3169, 1990.
42. D. Boudreau and P. Kabal, "Joint time delay estimation and adaptive recursive least squares filtering," *15th Biennial Symposium on Communications*, pp. 164-167, 1990.
43. H. L. Van Trees, *Detection, Estimation, and Modulation Theory-Part 1*, Wiley, 1968.
44. H. L. Van Trees, *Detection, Estimation, and Modulation Theory-Part 3*, Wiley, 1971.
45. A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1984.
46. H. Meyr, "Delay-lock tracking of stochastic signals," *IEEE Trans. on Communications*, vol. COM-24, pp. 331-339, March 1976.
47. S. Shaffer and C. S. Williams, "Comparison of the LMS, α LMS, and data reusing LMS algorithms," *18th Annual Asilomar Conference on Circuits, Systems, and Computers*, pp. 260-264, 1984.
48. A. Goldstein, *Constructive Real Analysis*, New York: Harper and Row, 1967.
49. P. Kabal, "The stability of adaptive minimum mean square error equalizers using delayed adjustment," *IEEE Trans. on Communications*, vol. COM-31, pp. 430-432, March 1983.
50. A. V. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, 1989.
51. L. G. Kelly, *Handbook of Numerical Methods and Applications*, Addison-Wesley, 1967.
52. L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. on Automatic Control*, vol. AC-22, pp. 551-575, August 1977.
53. G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control*, Prentice-Hall, 1984.
54. E. Mazo, "On the independence theory of equalizer convergence," *The Bell System Technical Journal*, vol. 58, pp. 963-993, May-June 1979.
55. Gilbert Strang, *Linear Algebra and Its Applications*, Academic Press, 1976.
56. G. Long, F. Ling and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 1397-1405, September 1989.
57. J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*, Macmillan Publishing Company, 1988.
58. B. C. Kuo, *Digital Control Systems*, SLR Publishing Company, 1977.
59. L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 722-736, June 1981.

60. A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 222-230, February 1985.
61. J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304-337, April 1984.
62. G. Carayannis, D. M. Manolakis and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394-1402, December 1983.
63. N. Kalouptsidis, G. Carayannis and D. G. Manolakis, "Fast design of multichannel FIR least-squares filters with optimum lag," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 48-59, February 1984.
64. W. Rudin, *Real and Complex Analysis*, 3rd edition, McGraw-Hill, 1987.
65. S. T. Alexander, "Fast adaptive filters: a geometrical approach," *IEEE ASSP Magazine*, pp. 18-28, October 1986.
66. K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*, John Wiley and Sons, 1988.
67. D. H. Youn, N. Ahmed and G. C. Carter, "A method for generating a class of time-delayed signals," *IEEE 1981 International Conference on Acoustic, Speech and Signal Processing*, pp. 1257-1260, 1981.
68. S. Ljung and L. Ljung, "Error propagation properties of recursive least-squares adaptation algorithms," *Automatica*, vol. 21, pp. 157-167, 1985.
69. D. W. Lin, "On digital implementation of the fast Kalman algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 998-1005, October 1984.
70. D. T. M. Slock and T. Kailath, "Numerically stable fast recursive least-squares transversal filters," *IEEE 1988 International Conference on Acoustic, Speech and Signal Processing*, pp. 1365-1368, 1988.
71. A. Benallal and A. Gilloire, "A new method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors," *IEEE 1988 International Conference on Acoustic, Speech and Signal Processing*, pp. 1373-1376, 1988.
72. J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of Acoustical Society of America*, vol. 65, pp. 943-950, April 1979.
73. F. Ling and J.G. Proakis, "Nonstationary learning characteristics of least squares adaptive estimation algorithms," *IEEE 1984 International Conference on Acoustic, Speech and Signal Processing*, pp. 3.7.1-3.7.4, 1984.
74. M. O'Flynn, *Probabilities, Random Variables, and Random Processes*, Harper and Row, 1982.