Subband Adaptive Filtering
Theory and Implementation

Kong-Aik Lee
Institute for Infocomm Research, Singapore

Woon-Seng Gan
Nanyang Technological University, Singapore

Sen M. Kuo
Northern Illinois University, USA
Subband Adaptive Filtering
Subband Adaptive Filtering
Theory and Implementation

Kong-Aik Lee
Institute for Infocomm Research, Singapore

Woon-Seng Gan
Nanyang Technological University, Singapore

Sen M. Kuo
Northern Illinois University, USA

WILEY
A John Wiley and Sons, Ltd., Publication
# Contents

About the authors xi  
Preface xiii  
Acknowledgments xv  
List of symbols xvii  
List of abbreviations xix  

1 Introduction to adaptive filters 1  
1.1 Adaptive filtering 1  
1.2 Adaptive transversal filters 2  
1.3 Performance surfaces 4  
1.4 Adaptive algorithms 6  
1.5 Spectral dynamic range and misadjustment 13  
1.6 Applications of adaptive filters 15  
1.6.1 Adaptive system identification 15  
1.6.2 Adaptive prediction 23  
1.6.3 Adaptive inverse modeling 25  
1.6.4 Adaptive array processing 28  
1.6.5 Summary of adaptive filtering applications 31  
1.7 Transform-domain and subband adaptive filters 31  
1.7.1 Transform-domain adaptive filters 31  
1.7.2 Subband adaptive filters 38  
1.8 Summary 39  
References 39  

2 Subband decomposition and multirate systems 41  
2.1 Multirate systems 41  
2.2 Filter banks 44  
2.2.1 Input–output relation 46
5 Critically sampled and oversampled subband structures 133
5.1 Variants of critically sampled subband adaptive filters 133
5.1.1 SAF with the affine projection algorithm 134
5.1.2 SAF with variable step sizes 136
5.1.3 SAF with selective coefficient update 137
5.2 Oversampled and nonuniform subband adaptive filters 138
5.2.1 Oversampled subband adaptive filtering 138
5.2.2 Nonuniform subband adaptive filtering 140
5.3 Filter bank design 141
5.3.1 Generalized DFT filter banks 141
5.3.2 Single-sideband modulation filter banks 142
5.3.3 Filter design criteria for DFT filter banks 144
5.3.4 Quadrature mirror filter banks 149
5.3.5 Pseudo-quadrature mirror filter banks 153
5.3.6 Conjugate quadrature filter banks 155
5.4 Case study: Proportionate subband adaptive filtering 156
5.4.1 Multiband structure with proportionate adaptation 156
5.4.2 MATLAB simulations 157
5.5 Summary 161
References 163

6 Multiband-structured subband adaptive filters 167
6.1 Multiband structure 167
6.1.1 Polyphase implementation 170
6.2 Multiband adaptation 173
6.2.1 Principle of minimal disturbance 173
6.2.2 Constrained subband updates 173
6.2.3 Computational complexity 175
6.3 Underdetermined least-squares solutions 177
6.3.1 NLMS equivalent 178
6.3.2 Projection interpretation 179
6.4 Stochastic interpretations 179
6.4.1 Stochastic approximation to Newton’s method 179
6.4.2 Weighted MSE criterion 181
6.4.3 Decorrelating properties 186
6.5 Filter bank design issues 187
6.5.1 The diagonal assumption 187
6.5.2 Power complementary filter bank 187
6.5.3 The number of subbands 188
6.6 Delayless MSAF 189
6.6.1 Open-loop configuration 189
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6.2</td>
<td>Closed-loop configuration</td>
<td>191</td>
</tr>
<tr>
<td>6.7</td>
<td>MATLAB examples</td>
<td>192</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Convergence of the MSAF algorithm</td>
<td>193</td>
</tr>
<tr>
<td>6.7.2</td>
<td>Subband and time-domain constraints</td>
<td>195</td>
</tr>
<tr>
<td>6.8</td>
<td>Summary</td>
<td>198</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>199</td>
</tr>
<tr>
<td>7</td>
<td>Stability and performance analysis</td>
<td>203</td>
</tr>
<tr>
<td>7.1</td>
<td>Algorithm, data model and assumptions</td>
<td>203</td>
</tr>
<tr>
<td>7.1.1</td>
<td>The MSAF algorithm</td>
<td>203</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Linear data model</td>
<td>204</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Paraunitary filter banks</td>
<td>206</td>
</tr>
<tr>
<td>7.2</td>
<td>Multiband MSE function</td>
<td>209</td>
</tr>
<tr>
<td>7.2.1</td>
<td>MSE functions</td>
<td>209</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Excess MSE</td>
<td>210</td>
</tr>
<tr>
<td>7.3</td>
<td>Mean analysis</td>
<td>211</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Projection interpretation</td>
<td>211</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Mean behavior</td>
<td>213</td>
</tr>
<tr>
<td>7.4</td>
<td>Mean-square analysis</td>
<td>214</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Energy conservation relation</td>
<td>214</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Variance relation</td>
<td>216</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Stability of the MSAF algorithm</td>
<td>216</td>
</tr>
<tr>
<td>7.4.4</td>
<td>Steady-state excess MSE</td>
<td>217</td>
</tr>
<tr>
<td>7.5</td>
<td>MATLAB examples</td>
<td>219</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Mean of the projection matrix</td>
<td>219</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Stability bounds</td>
<td>220</td>
</tr>
<tr>
<td>7.5.3</td>
<td>Steady-state excess MSE</td>
<td>222</td>
</tr>
<tr>
<td>7.6</td>
<td>Summary</td>
<td>223</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>224</td>
</tr>
<tr>
<td>8</td>
<td>New research directions</td>
<td>227</td>
</tr>
<tr>
<td>8.1</td>
<td>Recent research on filter bank design</td>
<td>227</td>
</tr>
<tr>
<td>8.2</td>
<td>New SAF structures and algorithms</td>
<td>228</td>
</tr>
<tr>
<td>8.2.1</td>
<td>In-band aliasing cancellation</td>
<td>228</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Adaptive algorithms for the SAF</td>
<td>230</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Variable tap lengths for the SAF</td>
<td>230</td>
</tr>
<tr>
<td>8.3</td>
<td>Theoretical analysis</td>
<td>232</td>
</tr>
<tr>
<td>8.4</td>
<td>Applications of the SAF</td>
<td>232</td>
</tr>
<tr>
<td>8.5</td>
<td>Further research on a multiband-structured SAF</td>
<td>233</td>
</tr>
<tr>
<td>8.6</td>
<td>Concluding remarks</td>
<td>234</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>235</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Programming in MATLAB</td>
<td>241</td>
</tr>
<tr>
<td>A.1</td>
<td>MATLAB fundamentals</td>
<td>241</td>
</tr>
<tr>
<td>A.1.1</td>
<td>Starting MATLAB</td>
<td>241</td>
</tr>
<tr>
<td>A.1.2</td>
<td>Constructing and manipulating matrices</td>
<td>244</td>
</tr>
<tr>
<td>A.1.3</td>
<td>The colon operator</td>
<td>244</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>ix</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>A.1.4 Data types</td>
<td>248</td>
<td></td>
</tr>
<tr>
<td>A.1.5 Working with strings</td>
<td>248</td>
<td></td>
</tr>
<tr>
<td>A.1.6 Cell arrays and structures</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td>A.1.7 MATLAB scripting with M-files</td>
<td>251</td>
<td></td>
</tr>
<tr>
<td>A.1.8 Plotting in MATLAB</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td>A.1.9 Other useful commands and tips</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>A.2 Signal processing toolbox</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>A.2.1 Quick fact about the signal processing toolbox</td>
<td>258</td>
<td></td>
</tr>
<tr>
<td>A.2.2 Signal processing tool</td>
<td>262</td>
<td></td>
</tr>
<tr>
<td>A.2.3 Window design and analysis tool</td>
<td>267</td>
<td></td>
</tr>
<tr>
<td>A.3 Filter design toolbox</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>A.3.1 Quick fact about the filter design toolbox</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>A.3.2 Filter design and analysis tool</td>
<td>269</td>
<td></td>
</tr>
<tr>
<td>A.3.3 MATLAB functions for adaptive filtering</td>
<td>270</td>
<td></td>
</tr>
<tr>
<td>A.3.4 A case study: adaptive noise cancellation</td>
<td>272</td>
<td></td>
</tr>
</tbody>
</table>

**Appendix B** Using MATLAB for adaptive filtering and subband adaptive filtering | 279
---
B.1 Digital signal processing | 279
B.1.1 Discrete-time signals and systems | 279
B.1.2 Signal representations in MATLAB | 280
B.2 Filtering and adaptive filtering in MATLAB | 282
B.2.1 FIR filtering | 282
B.2.2 The LMS adaptive algorithm | 284
B.2.3 Anatomy of the LMS code in MATLAB | 285
B.3 Multirate and subband adaptive filtering | 292
B.3.1 Implementation of multirate filter banks | 292
B.3.2 Implementation of a subband adaptive filter | 297

**Appendix C** Summary of MATLAB scripts, functions, examples and demos 301

**Appendix D** Complexity analysis of adaptive algorithms 307

Index 317
About the authors

Kong-Aik Lee received his B.Eng (1st Class Hons) degree from Universiti Teknologi Malaysia in 1999, and his Ph.D. degree from Nanyang Technological University, Singapore, in 2006. He is currently a Research Fellow with the Institute for Infocomm Research (I²R), Agency for Science, Technology and Research (A*STAR), Singapore. He has been actively involved in the research on subband adaptive filtering techniques for the past few years. He invented the Multiband-structured Subband Adaptive Filter (MSAF), a very fast converging and computationally efficient subband adaptive filtering algorithm. His current research has primarily focused on improved classifier design for speaker and language recognition.

Woon-Seng Gan received his B.Eng (1st Class Hons) and PhD degrees, both in Electrical and Electronic Engineering from the University of Strathclyde, UK in 1989 and 1993 respectively. He joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Lecturer and Senior Lecturer in 1993 and 1998 respectively. In 1999, he was promoted to Associate Professor. He is currently the Deputy Director of the Center for Signal Processing at Nanyang Technological University. His research interests include adaptive signal processing, psycho-acoustical signal processing, audio processing, and real-time embedded systems.

He has published more than 170 international refereed journals and conference papers, and has been awarded four Singapore and US patents. He has previously co-authored two technical books on Digital Signal Processors: Architectures, Implementations, and Applications (Prentice Hall, 2005) and Embedded Signal Processing with the Micro Signal Architecture (Wiley-IEEE, 2007).

Dr. Gan has also won the Institute of Engineers Singapore (IES) Prestigious Engineering Achievement Award in 2001 for his work on Audio Beam System. He is currently serving as an Associate Editor for the EURASIP Journal on Audio, Speech and Music Processing, and EURASIP Research Letters in Signal Processing. He is also a Senior Member of IEEE and serves as a committee member in the IEEE Signal Processing Society Education Technical Committee.

Sen M. Kuo received the B.S. degree from National Taiwan Normal University, Taipei, Taiwan, in 1976 and the M.S. and Ph.D. degrees from the University of New Mexico, Albuquerque, NM in 1983 and 1985, respectively.
He is a Professor and served as the department chair from 2002 to 2008 in the Department of Electrical Engineering, Northern Illinois University, DeKalb, IL. He was with Texas Instruments, Houston, TX in 1993, and with Chung-Ang University, Seoul, Korea in 2008. He is the leading author of four books: *Active Noise Control Systems* (Wiley, 1996), *Real-Time Digital Signal Processing* (Wiley, 2001, 2006), and *Digital Signal Processors* (Prentice Hall, 2005), and a co-author of *Embedded Signal Processing with the Micro Signal Architecture* (Wiley 2007). He holds seven US patents, and has published over 200 technical papers. His research focuses on active noise and vibration control, real-time DSP applications, adaptive echo and noise cancellation, digital audio and communication applications, and biomedical signal processing.

Prof. Kuo received the IEEE first-place transactions (Consumer Electronics) paper award in 1993, and the faculty-of-year award in 2001 for accomplishments in research and scholarly areas. He served as an associate editor for IEEE Transactions on Audio, Speech and Language Processing, and serves as a member of the editorial boards for EURASIP Research Letters in Signal Processing and Journal of Electrical and Computer Engineering.
Subband adaptive filtering is rapidly becoming one of the most effective techniques for reducing computational complexity and improving the convergence rate of algorithms in adaptive signal processing applications. Additional features of subband adaptive filters also make this technique suitable for many real-life applications.

This book covers the fundamental theory and analysis of commonly used subband adaptive filter structures and algorithms with concise derivations. The book is further enhanced with ready-to-run software written in MATLAB for researchers, engineers and students. These MATLAB codes are introduced in many chapters in order to clarify important concepts of subband adaptive algorithms, create figures and tables that will promote the understanding of theory, implement some important applications and provide a further extension of research works in this area. These programs along with associated data files and useful information are included in the companion CD.

This book serves as a state-of-the-art and comprehensive text/reference for researchers, practicing engineers and students who are developing and researching in advanced signal processing fields involving subband adaptive filters. The book consists of eight chapters, outlined as follows:

- **Chapter 1** covers fundamentals of adaptive filtering and introduces four basic applications. The general problems and difficulties encountered in adaptive filtering are presented and the motivations for using subband adaptive filters are explained.

- **Chapter 2** reviews basic concepts, design techniques and various formulations of multirate filter banks. In particular, we introduce a special class of paraunitary filter banks that can perfectly reconstruct the fullband signal after subband decomposition. The underlying affinity between filter bank and block transform is also addressed. Finally, techniques for designing cosine-modulated and discrete Fourier transform (DFT) filter banks that will be used extensively in the book are discussed.

- **Chapter 3** introduces important concepts of subband orthogonality and analyzes the correlation between subband signals. The second-order characteristics of multirate filter banks are discussed using a correlation-domain formulation. With this analysis technique, the effect of filtering random signals using a filter bank can be conveniently described in terms of the system effect on the autocorrelation function of the input signal.
Chapter 4 introduces fundamental principles of using subband adaptive filtering. The major problems and some recent improvements are also addressed. The delayless SAF structures, which eliminate the signal path delay introduced by the filter banks, are also presented in this chapter. Different delayless SAF structures and the corresponding weight transformation techniques are discussed.

Chapter 5 analyzes and compares various subband adaptive filter structures and techniques used to update tap weights in individual subbands. The main features of critically sampled and oversampled subband adaptive filter structures are also discussed. Each structure has its own filter bank design issues to be considered, which will be demonstrated using MATLAB examples.

Chapter 6 presents a class of multiband-structured subband adaptive filters that uses a set of normalized subband signals in order to adapt the fullband tap weights of a single adaptive filter. The fundamental idea and convergence behavior of the multiband weight-control mechanism is described. Filter bank design issues and delayless implementation for the multiband-structured subband adaptive filters are also discussed in this chapter.

Chapter 7 focuses on the stability and performance analysis of subband adaptive filters. Convergence analysis of subband adaptive filters in the mean and mean-square senses are presented. These derivations resulted in the stability bounds for step size and the steady-state MSE. The theoretical analysis provides a set of working rules for designing SAFs in practical applications.

Chapter 8 is the last chapter and introduces recent research works in the areas of subband adaptive filtering. In particular, new subband adaptive filter structures and algorithms, and the latest development in theoretical analysis and applications of subband adaptive filtering are discussed. This chapter also summarizes some research directions in multiband-structured subband adaptive filters.

As in many other technical reference books, we have attempted to cover all state-of-the-art techniques in subband adaptive filtering. However, there will still be some inevitable omissions due to the page limitation and publication deadline. A companion website (www3.ntu.edu.sg/home/ewsgan/saf_book.html) for this book has been created and will be continuously updated in order to provide references to the most recent progress in the area of subband adaptive filtering. Promising research topics and new applications will also be posted on this website so that researchers and students can become aware of challenging areas. We hope that this book will serve as a useful guide for what has already been done and as an inspiration for what will follow in the emerging area of subband adaptive filtering.
Acknowledgments

We are very grateful to many individuals for their assistance with writing this book. In particular, we would like to thank Kevin Kuo for proofreading the book, Dennis R. Morgan and Stephen Weiss for their patience in reviewing two early chapters of the book, as well as their helpful suggestions that have guided the presentation layout of the book. We would also like to thank Paulo S. R. Diniz and Behrouz Farhang-Boroujeny for valuable comments on an earlier version of the book.

Several individuals at John Wiley & Sons, Ltd provided great help in making this book a reality. We wish to thank Simone Taylor (Editor for Engineering Technology), Georgia Pinteau (Commissioning Editor), Liz Benson (Content Editor), Emily Bone (Associate Commissioning Editor) and Nicky Skinner (Project Editor) for their help in seeing this book through to the final stage.

This book is dedicated to many of our past and present students who have taken our digital signal processing and adaptive signal processing courses and have written MS theses and PhD dissertations and completed projects under our guidance at both Nanyang Technological University (NTU), Singapore, and Northern Illinois University (NIU), USA. Both institutions have provided us with a stimulating environment for research and teaching, and we appreciate the strong encouragement and support we have received. We would also like to mention the Institute for Infocomm Research (I²R), Singapore, for their support and encouragement in writing this book. Finally, we are greatly indebted to our parents and families for their understanding, patience and encouragement throughout this period.

Kong-Aik Lee, Woon-Seng Gan and Sen M. Kuo
November 2008
List of symbols

\( A^{-1} \)  Inverse of matrix \( A \)
\( A^T \)  Transposition of matrix \( A \)
\( A^H \)  Hermitian transposition of matrix \( A \)
\( \kappa(A) \)  Condition number of matrix \( A \)
\( I_{N \times N} \)  \( N \times N \) identity matrix
\( \Lambda = \text{diag}(\cdot) \)  Diagonal matrix
\( \tilde{E}(z) \)  Paraconjugate of \( E(z) \)
\( \| \cdot \| \)  Euclidean norm
\( | \cdot | \)  Absolute value
\( [\cdot] \)  Integer part of its argument
\( (\cdot)^* \)  Complex conjugation
\( E\{\cdot\} \)  Statistical expectation
\( * \)  Convolution
\( \delta(n) \)  Unit sample sequence
\( O(\cdot) \)  Order of complexity
\( \approx \)  Approximately equal to
\( \forall \)  For all
\( \infty \)  Infinity
\( \nabla \)  Gradient
\( \equiv \)  Defined as
\( j \)  \( \sqrt{-1} \)
\( \downarrow D \)  \( D \)-fold decimation
\( \uparrow I \)  \( I \)-fold interpolation
\( W_D \equiv \frac{\sqrt{e^{-j2\pi}}}{D} \)  \( D \)th root of unity
\( z^{-1} \)  Unit delay
\( z^{-\Delta} \)  \( \Delta \) samples delay
\( \gamma_{uu}(l) \)  Autocorrelation function of \( u(n) \)
\( \Gamma_{uu}(e^{j\omega}) \)  Power spectrum of \( u(n) \)
\( N \)  Number of subbands
\( M \)  Length of fullband adaptive filter
\( M_S \)  Length of subband adaptive filter
\( L \)  Length of analysis and synthesis filters
**LIST OF SYMBOLS**

- $P$: Project order of affine projection algorithm
- $p(n), P(z)$: Prototype filter
- $h_i(n), H_i(z)$: The $i$th analysis filter
- $f_i(n), F_i(z)$: The $i$th synthesis filter
- $H$: Analysis filter bank ($L \times N$ matrix)
- $F$: Synthesis filter bank ($L \times N$ matrix)
- $x_i(n), X_i(z)$: The $i$th subband signal
- $x_{i,D}(k), X_{i,D}(z)$: The $i$th subband signal after decimation
- $\gamma_{ip}(l)$: Cross-correlation function between the $i$th and $p$th subbands
- $\Gamma_{ip}(e^{j\omega})$: Cross-power spectrum between the $i$th and $p$th subbands
- $J = E\{e^2(n)\}$: MSE function
- $J_M$: Multiband MSE function
- $J_{LS}$: Weighted least-square error
- $J_{SB}$: Subband MSE function
- $J(\infty)$: Steady-state MSE
- $J_{es}$: Steady-state excess MSE
- $u(n)$: Input signal
- $d(n)$: Desired response
- $e(n)$: Error signal
- $\mu$: Step-size parameter
- $\alpha$: Regularization parameter
- $R$: Input autocorrelation matrix
- $p$: Cross-correlation vector
- $w_o$: Optimum weight vector
- $b, B(z)$: Unknown system
- $w(n), W(z)$: Adaptive weight vector (fullband)
- $w_i(k), W_i(z)$: Adaptive filter in the $i$th subband
List of abbreviations

ANC  Active noise control
AEC  Acoustic echo cancellation
AP   Affine projection
AR   Autoregressive
DCT  Discrete cosine transform
DFT  Discrete Fourier transform
e.g. exempli gratia: for example
FFT  Fast Fourier transform
FIR  Finite impulse response
IDFT Inverse DFT
i.e. id est: that is
IFFT Inverse FFT
IIR  Infinite impulse response
LMS  Least-mean-square
MSAF Multiband-structured subband adaptive filter
MSE  Mean-square error
NLMS Normalized LMS
PRA  Partial rank algorithm
PQMF Pseudo-QMF
QMF  Quadrature mirror filter
SAF  Subband adaptive filter
RLS  Recursive least-squares
Introduction to adaptive filters

This chapter introduces the fundamental principles of adaptive filtering and commonly used adaptive filter structures and algorithms. Basic concepts of applying adaptive filters in practical applications are also highlighted. The problems and difficulties encountered in time-domain adaptive filters are addressed. Subband adaptive filtering is introduced to solve these problems. Some adaptive filters are implemented using MATLAB for different applications. These ready-to-run programs can be modified to speed up research and development in adaptive signal processing.

1.1 Adaptive filtering

A filter is designed and used to extract or enhance the desired information contained in a signal. An adaptive filter is a filter with an associated adaptive algorithm for updating filter coefficients so that the filter can be operated in an unknown and changing environment. The adaptive algorithm determines filter characteristics by adjusting filter coefficients (or tap weights) according to the signal conditions and performance criteria (or quality assessment). A typical performance criterion is based on an error signal, which is the difference between the filter output signal and a given reference (or desired) signal.

As shown in Figure 1.1, an adaptive filter is a digital filter with coefficients that are determined and updated by an adaptive algorithm. Therefore, the adaptive algorithm behaves like a human operator that has the ability to adapt in a changing environment. For example, a human operator can avoid a collision by examining the visual information (input signal) based on his/her past experience (desired or reference signal) and by using visual guidance (performance feedback signal) to direct the vehicle to a safe position (output signal).

Adaptive filtering finds practical applications in many diverse fields such as communications, radar, sonar, control, navigation, seismology, biomedical engineering and even in financial engineering [1–7]. In Section 1.6, we will introduce some typical applications using adaptive filtering.

This chapter also briefly introduces subband adaptive filters that are more computationally efficient for applications that need a longer filter length, and are more effective if
the input signal is highly correlated. For example, high-order adaptive filters are required to cancel the acoustic echo in hands-free telephones [2]. The high-order filter together with a highly correlated input signal degrades the performances of most time-domain adaptive filters. Adaptive algorithms that are effective in dealing with ill-conditioning problems are available; however, such algorithms are usually computationally demanding, thereby limiting their use in many real-world applications.

In the following sections we introduce fundamental adaptive filtering concepts with an emphasis on widely used adaptive filter structures and algorithms. Several important properties on convergence rate and characteristics of input signals are also summarized. These problems motivated the development of subband adaptive filtering techniques in the forthcoming chapters. In-depth discussion on general adaptive signal processing can be found in many reference textbooks [1–3, 5–9].

1.2 Adaptive transversal filters

An adaptive filter is a self-designing and time-varying system that uses a recursive algorithm continuously to adjust its tap weights for operation in an unknown environment [6]. Figure 1.2 shows a typical structure of the adaptive filter, which consists of two basic functional blocks: (i) a digital filter to perform the desired filtering and (ii) an adaptive algorithm to adjust the tap weights of the filter. The digital filter computes the output $y(n)$ in response to the input signal $u(n)$, and generates an error signal $e(n)$ by comparing $y(n)$ with the desired response $d(n)$, which is also called the reference signal, as shown in Figure 1.1. The performance feedback signal $e(n)$ (also called the error signal) is used by the adaptive algorithm to adjust the tap weights of the digital filter.

The digital filter shown in Figure 1.2 can be realized using many different structures. The commonly used transversal or finite impulse response (FIR) filter is shown in
Figure 1.2 Typical structure of the adaptive filter using input and error signals to update its tap weights

Figure 1.3 An $M$-tap adaptive transversal filter

Figure 1.3. The adjustable tap weights, $w_m(n), m = 0, 1, \ldots, M - 1$, indicated by circles with arrows through them, are the filter tap weights at time instance $n$ and $M$ is the filter length. These time-varying tap weights form an $M \times 1$ weight vector expressed as

$$w(n) \equiv [w_0(n), w_1(n), \ldots, w_{M-1}(n)]^T,$$

where the superscript $T$ denotes the transpose operation of the matrix. Similarly, the input signal samples, $u(n - m), m = 0, 1, \ldots, M - 1$, form an $M \times 1$ input vector

$$u(n) \equiv [u(n), u(n - 1), \ldots, u(n - M + 1)]^T.$$

With these vectors, the output signal $y(n)$ of the adaptive FIR filter can be computed as the inner product of $w(n)$ and $u(n)$, expressed as

$$y(n) = \sum_{m=0}^{M-1} w_m(n)u(n - m) = w^T(n)u(n).$$
1.3 Performance surfaces

The error signal $e(n)$ shown in Figure 1.2 is the difference between the desired response $d(n)$ and the filter response $y(n)$, expressed as

$$e(n) = d(n) - w^T(n)u(n).$$  \hspace{1cm} (1.4)

The weight vector $w(n)$ is updated iteratively such that the error signal $e(n)$ is minimized. A commonly used performance criterion (or cost function) is the minimization of the mean-square error (MSE), which is defined as the expectation of the squared error as

$$J \equiv E \{e^2(n)\}. \hspace{1cm} (1.5)$$

For a given weight vector $w = [w_0, w_1, \ldots, w_{M-1}]^T$ with stationary input signal $u(n)$ and desired response $d(n)$, the MSE can be calculated from Equations (1.4) and (1.5) as

$$J = E \{d^2(n)\} - 2p^T w + w^T Rw,$$  \hspace{1cm} (1.6)

where $R \equiv E\{u(n)u^T(n)\}$ is the input autocorrelation matrix and $p \equiv E\{d(n)u(n)\}$ is the cross-correlation vector between the desired response and the input vector. The time index $n$ has been dropped in Equation (1.6) from the vector $w(n)$ because the MSE is treated as a stationary function.

Equation (1.6) shows that the MSE is a quadratic function of the tap weights $\{w_0, w_1, \ldots, w_{M-1}\}$ since they appear in first and second degrees only. A typical performance (or error) surface for a two-tap transversal filter is shown in Figure 1.4. For $M > 2$, the error surface is a hyperboloid. The quadratic performance surface guarantees that it has a single global minimum MSE corresponding to the

![Figure 1.4](image.png)  

Figure 1.4  A typical performance surface for a two-tap adaptive FIR filter
optimum vector \( \mathbf{w}_o \). The optimum solution can be obtained by taking the first derivative of Equation (1.6) with respect to \( \mathbf{w} \) and setting the derivative to zero. This results in the Wiener–Hopf equation

\[
\mathbf{Rw}_o = \mathbf{p}. \tag{1.7}
\]

Assuming that \( \mathbf{R} \) has an inverse matrix, the optimum weight vector is

\[
\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}. \tag{1.8}
\]

Substituting Equation (1.8) into (1.6), the minimum MSE corresponding to the optimum weight vector can be obtained as

\[
J_{\text{min}} = E\{d^2(n)\} - \mathbf{p}^T \mathbf{w}_o. \tag{1.9}
\]

**Example 1.1** (a complete listing is given in the M-file `Example_1P1.m`)

Consider a two-tap case for the FIR filter, as shown in Figure 1.3. Assuming that the input signal and desired response are stationary with the second-order statistics

\[
\mathbf{R} = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} \quad \text{and} \quad E\{d^2(n)\} = 3.
\]

The performance surface (depicted in Figure 1.4) is defined by Equation (1.6) and can be constructed by computing the MSE at different values of \( w_0 \) and \( w_1 \), as demonstrated by the following partial listing:

```matlab
R = [1.1 0.5; 0.5 1.1]; % Input autocorrelation matrix
p = [0.5272; -0.4458]; % Cross-correlation vector
dn_var = 3; % Variance of desired response d(n)

w0 = 0:0.1:4; % Range of first tap weight values
w1 = -4:0.1:0; % Range of second tap weight values
J = zeros(length(w0),length(w1)); % Clear MSE values at all (w0, w1)

for m = 1:length(w0)
    for n = 1:length(w1)
        w = [w0(m) w1(n)]';
        J(m,n) = dn_var-2*p'*w + w'*R*w; % Compute MSE values at all (w0, w1)
    end
end % points, store in J matrix

figure; meshc(w0,w1,J'); % Plot combination of mesh and contour
view([-130 22]); % Set the angle to view 3-D plot
hold on; w_opt = inv(R)*p;
plot(w_opt(1),w_opt(2),'o');
```
plot(w0,w_opt(2)*ones(length(w0),1));
plot(w_opt(1)*ones(length(w1),1),w1);
xlabel('w_0'); ylabel('w_1');
plot(w0,-4*ones(length(w0),1)); plot(4*ones(length(w1),1),w1);

1.4 Adaptive algorithms

An adaptive algorithm is a set of recursive equations used to adjust the weight vector $w(n)$ automatically to minimize the error signal $e(n)$ such that the weight vector converges iteratively to the optimum solution $w_0$ that corresponds to the bottom of the performance surface, i.e. the minimum MSE $J_{\text{min}}$. The least-mean-square (LMS) algorithm is the most widely used among various adaptive algorithms because of its simplicity and robustness. The LMS algorithm based on the steepest-descent method using the negative gradient of the instantaneous squared error, i.e. $J \approx e^2(n)$, was devised by Widrow and Stearns [6] to study the pattern-recognition machine. The LMS algorithm updates the weight vector as follows:

$$w(n + 1) = w(n) + \mu U(n)e(n),$$

where $\mu$ is the step size (or convergence factor) that determines the stability and the convergence rate of the algorithm. The function M-files of implementing the LMS algorithm are LMSinit.m and LMSadapt.m, and the script M-file that calls these MATLAB functions for demonstration of the LMS algorithm is LMSdemo.m. Partial listing of these files is shown below. The complete function M-files of the adaptive LMS algorithm and other adaptive algorithms discussed in this chapter are available in the companion CD.

The LMS algorithm

There are two MATLAB functions used to implement the adaptive FIR filter with the LMS algorithm. The first function LMSinit performs the initialization of the LMS algorithm, where the adaptive weight vector $w_0$ is normally clear to a zero vector at the start of simulation. The step size $\mu$ and the leaky factor $\text{leak}$ can be determined by the user at run time (see the complete LMSdemo.m for details). The second function LMSadapt performs the actual computation of the LMS algorithm given in Equation (1.10), where $u_n$ and $d_n$ are the input and desired signal vectors, respectively. $S$ is the initial state of the LMS algorithm, which is obtained from the output argument of the first function. The output arguments of the second function consist of the adaptive filter output sequence $y_n$, the error sequence $e_n$ and the final filter state $S$.

$$S = \text{LMSinit}(\text{zeros}(M,1),\mu); \quad \% \text{Initialization}$$
$$S,\text{unknownsys} = b;$$
[yn, en, S] = LMSadapt(un, dn, S); % Perform LMS algorithm
...

function S = LMSinit(w0, mu, leak)
% Assign structure fields
S.coeffs = w0(:); % Weight (column) vector of filter
S.step = mu; % Step size of LMS algorithm
S.leakage = leak; % Leaky factor for leaky LMS
S.iter = 0; % Iteration count
S.AdaptStart = length(w0); % Running effect of adaptive
% filter, minimum M

function [yn, en, S] = LMSadapt(un, dn, S)
M = length(S.coeffs); % Length of FIR filter
mu = S.step; % Step size of LMS algorithm
leak = S.leakage; % Leaky factor
AdaptStart = S.AdaptStart;
w = S.coeffs; % Weight vector of FIR filter
u = zeros(M, 1); % Input signal vector

% ITER = length(un); % Length of input sequence
yn = zeros(1, ITER); % Initialize output vector to zero
en = zeros(1, ITER); % Initialize error vector to zero

for n = 1:ITER
    u = [un(n); u(1:end-1)]; % Input signal vector contains
    % [u(n), u(n-1), ..., u(n-M+1)]'
    yn(n) = w'*u; % Output signal
    en(n) = dn(n) - yn(n); % Estimation error
    if ComputeEML == 1;
        eml(n) = norm(b-w)/norm_b; % System error norm (normalized)
    end
    if n >= AdaptStart
        w = (1-mu*leak)*w + (mu*en(n))*u; % Tap-weight adaptation
        S.iter = S.iter + 1;
    end
end

As shown in Equation (1.10), the LMS algorithm uses an iterative approach to adapt the tap weights to the optimum Wiener–Hopf solution given in Equation (1.8). To guarantee the stability of the algorithm, the step size is chosen in the range

\[ 0 < \mu < \frac{2}{\lambda_{\text{max}}}, \]  

(1.11)

where \( \lambda_{\text{max}} \) is the largest eigenvalue of the input autocorrelation matrix \( \mathbf{R} \). However, the eigenvalues of \( \mathbf{R} \) are usually not known in practice so the sum of the eigenvalues
(or the trace of $R$) is used to replace $\lambda_{\text{max}}$. Therefore, the step size is in the range of $0 < \mu < 2/\text{trace}(R)$. Since $\text{trace}(R) = MP_u$ is related to the average power $P_u$ of the input signal $u(n)$, a commonly used step size bound is obtained as

$$0 < \mu < \frac{2}{MP_u}.$$  \hspace{1cm} (1.12)

It has been shown that the stability of the algorithm requires a more stringent condition on the upper bound of $\mu$ when convergence of the weight variance is imposed [3]. For Gaussian signals, convergence of the MSE requires $0 < \mu < 2/3MP_u$.

The upper bound on $\mu$ provides an important guide in the selection of a suitable step size for the LMS algorithm. As shown in (1.12), a smaller step size $\mu$ is used to prevent instability for a larger filter length $M$. Also, the step size is inversely proportional to the input signal power $P_u$. Therefore, a stronger signal must use a smaller step size, while a weaker signal can use a larger step size. This relationship can be incorporated into the LMS algorithm by normalizing the step size with respect to the input signal power. This normalization of step size (or input signal) leads to a useful variant of the LMS algorithm known as the normalized LMS (NLMS) algorithm.

The NLMS algorithm [3] includes an additional normalization term $u^T(n)u(n)$, as shown in the following equation:

$$w(n+1) = w(n) + \mu \frac{u(n)}{u^T(n)u(n)}e(n),$$  \hspace{1cm} (1.13)

where the step size is now bounded in the range $0 < \mu < 2$. It makes the convergence rate independent of signal power by normalizing the input vector $u(n)$ with the energy $u^T(n)u(n) = \sum_{m=0}^{M-1} u^2(n - m)$ of the input signal in the adaptive filter. There is no significant difference between the convergence performance of the LMS and NLMS algorithms for stationary signals when the step size $\mu$ of the LMS algorithm is properly chosen. The advantage of the NLMS algorithm only becomes apparent for nonstationary signals like speech, where significantly faster convergence can be achieved for the same level of MSE in the steady state after the algorithm has converged. The function M-files for the NLMS algorithm are NLMSadapt.m and NLMSadapt.m. The script M-file that calls these MATLAB functions for demonstration is NLMSdemo.m. Partial listing of these files is shown below.

---

The NLMS algorithm

Most parameters of the NLMS algorithm are the same as the LMS algorithm, except that the step size $\mu$ is now bounded between 0 and 2. The normalization term, $u^T(n)u + \alpha$, makes the convergence rate independent of signal power. An additional constant $\alpha$ is used to avoid division by zero in normalization operations.

```matlab
S = NLMSinit(zeros(M,1),mu); % Initialization
S.unknownsys = b;
```