

Saeed V. Vaseghi

Advanced
Digital Signal
Processing
and
Noise
Reduction

Third Edition

 WILEY

TLFeBOOK

Advanced Digital Signal Processing and Noise Reduction

Advanced Digital Signal Processing and Noise Reduction

Third Edition

Saeed V. Vaseghi

*Professor of Communications and Signal Processing
Department of Electronics and Computer Engineering
Brunel University, UK*



John Wiley & Sons, Ltd

Copyright © 2006 John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester,
West Sussex PO19 8SQ, England

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk

Visit our Home Page on www.wiley.com

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except under the terms of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London W1T 4LP, UK, without the permission in writing of the Publisher. Requests to the Publisher should be addressed to the Permissions Department, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, or emailed to permreq@wiley.co.uk, or faxed to (+44) 1243 770620.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The Publisher is not associated with any product or vendor mentioned in this book.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Cataloging in Publication Data

Vaseghi, Saeed V.

Advanced digital signal processing and noise reduction / Saeed V. Vaseghi. — 3rd ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-470-09494-X

1. Signal processing. 2. Electronic noise. 3. Digital filters (Mathematics) I. Title.

TK5102.9.V37 2005

621.382'2—dc22

2005018514

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN-13 978-0-470-09494-5 (HB)

ISBN-10 0-470-09494-X (HB)

Typeset in 10/12pt Times by Integra Software Services Pvt. Ltd, Pondicherry, India.

Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.

This book is printed on acid-free paper responsibly manufactured from sustainable forestry in which at least two trees are planted for each one used for paper production.

To my Luke

I wish to thank Esfandiar Zavarehei and Wendy Pillar for proof reading this edition and for many excellent suggestions. Thanks also to Ben Milner, Qin Yan, Dimitrios Rentzo, Charles Ho and Aimin Chen.

Contents

Preface	xvii
Symbols	xxi
Abbreviations	xxv
1 Introduction	1
1.1 Signals and Information	1
1.2 Signal Processing Methods	3
1.2.1 Transform-based Signal Processing	3
1.2.2 Model-based Signal Processing	4
1.2.3 Bayesian Signal Processing	4
1.2.4 Neural Networks	5
1.3 Applications of Digital Signal Processing	5
1.3.1 Adaptive Noise Cancellation	5
1.3.2 Adaptive Noise Reduction	6
1.3.3 Blind Channel Equalisation	7
1.3.4 Signal Classification and Pattern Recognition	8
1.3.5 Linear Prediction Modelling of Speech	9
1.3.6 Digital Coding of Audio Signals	10
1.3.7 Detection of Signals in Noise	12
1.3.8 Directional Reception of Waves: Beam-forming	13
1.3.9 Dolby Noise Reduction	15
1.3.10 Radar Signal Processing: Doppler Frequency Shift	15
1.4 Sampling and Analogue-to-digital Conversion	17
1.4.1 Sampling and Reconstruction of Analogue Signals	18
1.4.2 Quantisation	19
Bibliography	21

2	Noise and Distortion	23
2.1	Introduction	24
2.2	White Noise	25
	2.2.1 Band-limited White Noise	26
2.3	Coloured Noise	26
2.4	Impulsive Noise	27
2.5	Transient Noise Pulses	29
2.6	Thermal Noise	30
2.7	Shot Noise	31
2.8	Electromagnetic Noise	31
2.9	Channel Distortions	32
2.10	Echo and Multipath Reflections	33
2.11	Modelling Noise	33
	2.11.1 Additive White Gaussian Noise Model	36
	2.11.2 Hidden Markov Model for Noise	36
	Bibliography	37
3	Probability and Information Models	39
3.1	Introduction	40
3.2	Random Signals	41
	3.2.1 Random and Stochastic Processes	43
	3.2.2 The Space of a Random Process	43
3.3	Probability Models	44
	3.3.1 Probability and Random Variables	45
	3.3.2 Probability Mass Function	45
	3.3.3 Probability Density Function	47
	3.3.4 Probability Density Functions of Random Processes	48
3.4	Information Models	50
	3.4.1 Entropy	51
	3.4.2 Mutual Information	54
	3.4.3 Entropy Coding	56
3.5	Stationary and Nonstationary Random Processes	59
	3.5.1 Strict-sense Stationary Processes	61
	3.5.2 Wide-sense Stationary Processes	61
	3.5.3 Nonstationary Processes	62
3.6	Statistics (Expected Values) of a Random Process	62
	3.6.1 The Mean Value	63
	3.6.2 Autocorrelation	63
	3.6.3 Autocovariance	66
	3.6.4 Power Spectral Density	66
	3.6.5 Joint Statistical Averages of Two Random Processes	68
	3.6.6 Cross-correlation and Cross-covariance	68
	3.6.7 Cross-power Spectral Density and Coherence	70
	3.6.8 Ergodic Processes and Time-averaged Statistics	70
	3.6.9 Mean-ergodic Processes	70
	3.6.10 Correlation-ergodic Processes	72

3.7	Some Useful Classes of Random Processes	73
3.7.1	Gaussian (Normal) Process	73
3.7.2	Multivariate Gaussian Process	74
3.7.3	Mixture Gaussian Process	75
3.7.4	A Binary-state Gaussian Process	76
3.7.5	Poisson Process	77
3.7.6	Shot Noise	78
3.7.7	Poisson–Gaussian Model for Clutters and Impulsive Noise	79
3.7.8	Markov Processes	80
3.7.9	Markov Chain Processes	81
3.7.10	Gamma Probability Distribution	82
3.7.11	Rayleigh Probability Distribution	83
3.7.12	Laplacian Probability Distribution	83
3.8	Transformation of a Random Process	83
3.8.1	Monotonic Transformation of Random Processes	84
3.8.2	Many-to-one Mapping of Random Signals	86
3.9	Summary	90
	Bibliography	90
4	Bayesian Inference	93
4.1	Bayesian Estimation Theory: Basic Definitions	94
4.1.1	Dynamic and Probability Models in Estimation	95
4.1.2	Parameter Space and Signal Space	96
4.1.3	Parameter Estimation and Signal Restoration	97
4.1.4	Performance Measures and Desirable Properties of Estimators	98
4.1.5	Prior and Posterior Spaces and Distributions	100
4.2	Bayesian Estimation	102
4.2.1	Maximum <i>a Posteriori</i> Estimation	103
4.2.2	Maximum-likelihood Estimation	104
4.2.3	Minimum Mean Square Error Estimation	107
4.2.4	Minimum Mean Absolute Value of Error Estimation	108
4.2.5	Equivalence of the MAP, ML, MMSE and MAVE for Gaussian Processes with Uniform Distributed Parameters	109
4.2.6	The Influence of the Prior on Estimation Bias and Variance	109
4.2.7	The Relative Importance of the Prior and the Observation	114
4.3	The Estimate–Maximise Method	116
4.3.1	Convergence of the EM Algorithm	117
4.4	Cramer–Rao Bound on the Minimum Estimator Variance	119
4.4.1	Cramer–Rao Bound for Random Parameters	120
4.4.2	Cramer–Rao Bound for a Vector Parameter	121
4.5	Design of Gaussian Mixture Models	121
4.5.1	EM Estimation of Gaussian Mixture Model	122
4.6	Bayesian Classification	124
4.6.1	Binary Classification	125
4.6.2	Classification Error	127
4.6.3	Bayesian Classification of Discrete-valued Parameters	128

4.6.4	Maximum <i>a Posteriori</i> Classification	128
4.6.5	Maximum-likelihood Classification	129
4.6.6	Minimum Mean Square Error Classification	129
4.6.7	Bayesian Classification of Finite State Processes	130
4.6.8	Bayesian Estimation of the Most Likely State Sequence	131
4.7	Modelling the Space of a Random Process	132
4.7.1	Vector Quantisation of a Random Process	132
4.7.2	Vector Quantisation using Gaussian Models	133
4.7.3	Design of a Vector Quantiser: <i>K</i> -means Clustering	133
4.8	Summary	134
	Bibliography	135
5	Hidden Markov Models	137
5.1	Statistical Models for Nonstationary Processes	138
5.2	Hidden Markov Models	139
5.2.1	Comparison of Markov and Hidden Markov Models	139
5.2.2	A Physical Interpretation: HMMs of Speech	141
5.2.3	Hidden Markov Model as a Bayesian Model	142
5.2.4	Parameters of a Hidden Markov Model	143
5.2.5	State Observation Probability Models	143
5.2.6	State Transition Probabilities	144
5.2.7	State–Time Trellis Diagram	145
5.3	Training Hidden Markov Models	145
5.3.1	Forward–Backward Probability Computation	147
5.3.2	Baum–Welch Model Re-estimation	148
5.3.3	Training HMMs with Discrete Density Observation Models	149
5.3.4	HMMs with Continuous Density Observation Models	150
5.3.5	HMMs with Gaussian Mixture pdfs	151
5.4	Decoding of Signals using Hidden Markov Models	152
5.4.1	Viterbi Decoding Algorithm	154
5.5	HMMs in DNA and Protein Sequence Modelling	155
5.6	HMMs for Modelling Speech and Noise	156
5.6.1	Modelling Speech with HMMs	156
5.6.2	HMM-based Estimation of Signals in Noise	156
5.6.3	Signal and Noise Model Combination and Decomposition	158
5.6.4	Hidden Markov Model Combination	159
5.6.5	Decomposition of State Sequences of Signal and Noise	160
5.6.6	HMM-based Wiener Filters	160
5.6.7	Modelling Noise Characteristics	162
5.7	Summary	162
	Bibliography	163
6	Least Square Error Filters	165
6.1	Least Square Error Estimation: Wiener Filters	166
6.2	Block-data Formulation of the Wiener Filter	170
6.2.1	QR Decomposition of the Least Square Error Equation	171

6.3	Interpretation of Wiener Filters as Projections in Vector Space	172
6.4	Analysis of the Least Mean Square Error Signal	174
6.5	Formulation of Wiener Filters in the Frequency Domain	175
6.6	Some Applications of Wiener Filters	177
6.6.1	Wiener Filters for Additive Noise Reduction	177
6.6.2	Wiener Filters and Separability of Signal and Noise	178
6.6.3	The Square-root Wiener Filter	179
6.6.4	Wiener Channel Equaliser	180
6.6.5	Time-alignment of Signals in Multichannel/Multisensor Systems	181
6.7	Implementation of Wiener Filters	182
6.7.1	The Choice of Wiener Filter Order	183
6.7.2	Improvements to Wiener Filters	184
6.8	Summary	185
	Bibliography	185
7	Adaptive Filters	187
7.1	Introduction	188
7.2	State-space Kalman Filters	188
7.2.1	Derivation of the Kalman Filter Algorithm	190
7.3	Sample-adaptive Filters	195
7.4	Recursive Least Square Adaptive Filters	196
7.4.1	The Matrix Inversion Lemma	198
7.4.2	Recursive Time-update of Filter Coefficients	199
7.5	The Steepest-descent Method	201
7.5.1	Convergence Rate	203
7.5.2	Vector-valued Adaptation Step Size	204
7.6	The LMS Filter	204
7.6.1	Leaky LMS Algorithm	205
7.6.2	Normalised LMS Algorithm	206
7.7	Summary	207
	Bibliography	208
8	Linear Prediction Models	209
8.1	Linear Prediction Coding	210
8.1.1	Frequency Response of LP Models	213
8.1.2	Calculation of Predictor Coefficients	214
8.1.3	Effect of Estimation of Correlation Function on LP Model Solution	216
8.1.4	The Inverse Filter: Spectral Whitening	216
8.1.5	The Prediction Error Signal	217
8.2	Forward, Backward and Lattice Predictors	219
8.2.1	Augmented Equations for Forward and Backward Predictors	220
8.2.2	Levinson–Durbin Recursive Solution	221
8.2.3	Lattice Predictors	223
8.2.4	Alternative Formulations of Least Square Error Prediction	224
8.2.5	Predictor Model Order Selection	225
8.3	Short- and Long-term Predictors	226

8.4	MAP Estimation of Predictor Coefficients	228
8.4.1	Probability Density Function of Predictor Output	229
8.4.2	Using the Prior pdf of the Predictor Coefficients	230
8.5	Formant-tracking LP Models	230
8.6	Sub-band Linear Prediction Model	232
8.7	Signal Restoration using Linear Prediction Models	233
8.7.1	Frequency-domain Signal Restoration using Prediction Models	235
8.7.2	Implementation of Sub-band Linear Prediction Wiener Filters	237
8.8	Summary	238
	Bibliography	238
9	Power Spectrum and Correlation	241
9.1	Power Spectrum and Correlation	242
9.2	Fourier Series: Representation of Periodic Signals	243
9.3	Fourier Transform: Representation of Aperiodic Signals	245
9.3.1	Discrete Fourier Transform	246
9.3.2	Time/Frequency Resolutions, the Uncertainty Principle	247
9.3.3	Energy-spectral Density and Power-spectral Density	248
9.4	Nonparametric Power Spectrum Estimation	249
9.4.1	The Mean and Variance of Periodograms	250
9.4.2	Averaging Periodograms (Bartlett Method)	250
9.4.3	Welch Method: Averaging Periodograms from Overlapped and Windowed Segments	251
9.4.4	Blackman–Tukey Method	252
9.4.5	Power Spectrum Estimation from Autocorrelation of Overlapped Segments	253
9.5	Model-based Power Spectrum Estimation	254
9.5.1	Maximum-entropy Spectral Estimation	255
9.5.2	Autoregressive Power Spectrum Estimation	257
9.5.3	Moving-average Power Spectrum Estimation	257
9.5.4	Autoregressive Moving-average Power Spectrum Estimation	258
9.6	High-resolution Spectral Estimation Based on Subspace Eigenanalysis	259
9.6.1	Pisarenko Harmonic Decomposition	259
9.6.2	Multiple Signal Classification Spectral Estimation	261
9.6.3	Estimation of Signal Parameters via Rotational Invariance Techniques	264
9.7	Summary	265
	Bibliography	266
10	Interpolation	267
10.1	Introduction	268
10.1.1	Interpolation of a Sampled Signal	268
10.1.2	Digital Interpolation by a Factor of I	269
10.1.3	Interpolation of a Sequence of Lost Samples	271
10.1.4	The Factors that affect Interpolation Accuracy	273

10.2	Polynomial Interpolation	274
10.2.1	Lagrange Polynomial Interpolation	275
10.2.2	Newton Polynomial Interpolation	276
10.2.3	Hermite Polynomial Interpolation	278
10.2.4	Cubic Spline Interpolation	278
10.3	Model-based Interpolation	280
10.3.1	Maximum <i>a Posteriori</i> Interpolation	281
10.3.2	Least Square Error Autoregressive Interpolation	282
10.3.3	Interpolation based on a Short-term Prediction Model	283
10.3.4	Interpolation based on Long- and Short-term Correlations	286
10.3.5	LSAR Interpolation Error	289
10.3.6	Interpolation in Frequency–Time Domain	290
10.3.7	Interpolation using Adaptive Codebooks	293
10.3.8	Interpolation through Signal Substitution	294
10.4	Summary	294
	Bibliography	295
11	Spectral Amplitude Estimation	297
11.1	Introduction	298
11.1.1	Spectral Representation of Noisy Signals	299
11.1.2	Vector Representation of the Spectrum of Noisy Signals	299
11.2	Spectral Subtraction	300
11.2.1	Power Spectrum Subtraction	302
11.2.2	Magnitude Spectrum Subtraction	303
11.2.3	Spectral Subtraction Filter: Relation to Wiener Filters	303
11.2.4	Processing Distortions	304
11.2.5	Effect of Spectral Subtraction on Signal Distribution	305
11.2.6	Reducing the Noise Variance	306
11.2.7	Filtering Out the Processing Distortions	307
11.2.8	Nonlinear Spectral Subtraction	308
11.2.9	Implementation of Spectral Subtraction	310
11.3	Bayesian MMSE Spectral Amplitude Estimation	312
11.4	Application to Speech Restoration and Recognition	315
11.5	Summary	315
	Bibliography	316
12	Impulsive Noise	319
12.1	Impulsive Noise	320
12.1.1	Autocorrelation and Power Spectrum of Impulsive Noise	322
12.2	Statistical Models for Impulsive Noise	323
12.2.1	Bernoulli–Gaussian Model of Impulsive Noise	324
12.2.2	Poisson–Gaussian Model of Impulsive Noise	324
12.2.3	A Binary-state Model of Impulsive Noise	325
12.2.4	Signal-to-impulsive-noise Ratio	326

12.3	Median Filters	327
12.4	Impulsive Noise Removal using Linear Prediction Models	328
12.4.1	Impulsive Noise Detection	328
12.4.2	Analysis of Improvement in Noise Detectability	330
12.4.3	Two-sided Predictor for Impulsive Noise Detection	331
12.4.4	Interpolation of Discarded Samples	332
12.5	Robust Parameter Estimation	333
12.6	Restoration of Archived Gramophone Records	334
12.7	Summary	335
	Bibliography	336
13	Transient Noise Pulses	337
13.1	Transient Noise Waveforms	337
13.2	Transient Noise Pulse Models	339
13.2.1	Noise Pulse Templates	340
13.2.2	Autoregressive Model of Transient Noise Pulses	341
13.2.3	Hidden Markov Model of a Noise Pulse Process	342
13.3	Detection of Noise Pulses	342
13.3.1	Matched Filter for Noise Pulse Detection	343
13.3.2	Noise Detection based on Inverse Filtering	344
13.3.3	Noise Detection based on HMM	344
13.4	Removal of Noise Pulse Distortions	345
13.4.1	Adaptive Subtraction of Noise Pulses	345
13.4.2	AR-based Restoration of Signals Distorted by Noise Pulses	347
13.5	Summary	349
	Bibliography	349
14	Echo Cancellation	351
14.1	Introduction: Acoustic and Hybrid Echoes	352
14.2	Telephone Line Hybrid Echo	353
14.2.1	Echo: the Sources of Delay in Telephone Networks	354
14.2.2	Echo Return Loss	355
14.3	Hybrid Echo Suppression	355
14.4	Adaptive Echo Cancellation	356
14.4.1	Echo Canceller Adaptation Methods	357
14.4.2	Convergence of Line Echo Canceller	358
14.4.3	Echo Cancellation for Digital Data Transmission	359
14.5	Acoustic Echo	360
14.6	Sub-band Acoustic Echo Cancellation	363
14.7	Multiple-input Multiple-output Echo Cancellation	365
14.7.1	Stereophonic Echo Cancellation Systems	365
14.8	Summary	368
	Bibliography	368

15	Channel Equalisation and Blind Deconvolution	371
15.1	Introduction	372
15.1.1	The Ideal Inverse Channel Filter	373
15.1.2	Equalisation Error, Convolutional Noise	374
15.1.3	Blind Equalisation	374
15.1.4	Minimum- and Maximum-phase Channels	376
15.1.5	Wiener Equaliser	377
15.2	Blind Equalisation using the Channel Input Power Spectrum	379
15.2.1	Homomorphic Equalisation	380
15.2.2	Homomorphic Equalisation using a Bank of High-pass Filters	382
15.3	Equalisation based on Linear Prediction Models	382
15.3.1	Blind Equalisation through Model Factorisation	384
15.4	Bayesian Blind Deconvolution and Equalisation	385
15.4.1	Conditional Mean Channel Estimation	386
15.4.2	Maximum-likelihood Channel Estimation	386
15.4.3	Maximum <i>a Posteriori</i> Channel Estimation	386
15.4.4	Channel Equalisation based on Hidden Markov Models	387
15.4.5	MAP Channel Estimate based on HMMs	389
15.4.6	Implementations of HMM-based Deconvolution	390
15.5	Blind Equalisation for Digital Communications Channels	393
15.5.1	LMS Blind Equalisation	395
15.5.2	Equalisation of a Binary Digital Channel	397
15.6	Equalisation based on Higher-order Statistics	398
15.6.1	Higher-order Moments, Cumulants and Spectra	399
15.6.2	Higher-order Spectra of Linear Time-invariant Systems	401
15.6.3	Blind Equalisation based on Higher-order Cepstra	402
15.7	Summary	406
	Bibliography	406
16	Speech Enhancement in Noise	409
16.1	Introduction	410
16.2	Single-input Speech-enhancement Methods	411
16.2.1	An Overview of a Speech-enhancement System	411
16.2.2	Wiener Filter for De-noising Speech	414
16.2.3	Spectral Subtraction of Noise	417
16.2.4	Bayesian MMSE Speech Enhancement	418
16.2.5	Kalman Filter	419
16.2.6	Speech Enhancement via LP Model Reconstruction	422
16.3	Multiple-input Speech-enhancement Methods	425
16.3.1	Beam-forming with Microphone Arrays	427
16.4	Speech Distortion Measurements	430
	Bibliography	431
17	Noise in Wireless Communications	433
17.1	Introduction to Cellular Communications	434
17.2	Noise, Capacity and Spectral Efficiency	436

17.3	Communications Signal Processing in Mobile Systems	438
17.4	Noise and Distortion in Mobile Communications Systems	439
17.4.1	Multipath Propagation of Electromagnetic Signals	440
17.4.2	Rake Receivers for Multipath Signals	441
17.4.3	Signal Fading in Mobile Communications Systems	442
17.4.4	Large-scale Signal Fading	443
17.4.5	Small-scale Fast Signal Fading	444
17.5	Smart Antennas	444
17.5.1	Switched and Adaptive Smart Antennas	446
17.5.2	Space–Time Signal Processing – Diversity Schemes	446
17.6	Summary	447
	Bibliography	448
Index		449

Preface

The applications of DSP are numerous and include multimedia technology, audio signal processing, video signal processing, cellular mobile communication, adaptive network management, radar systems, pattern analysis, pattern recognition, medical signal processing, financial data forecasting, artificial intelligence, decision making systems, control systems and information search engines.

The theory and application of signal processing is concerned with the identification, modelling and utilisation of patterns and structures in a signal process. The observation signals are often distorted, incomplete and noisy. Hence, noise reduction and the removal of channel distortion and interference are important parts of a signal processing system.

Since the publication of the first edition of this book in 1996, digital signal processing (DSP) in general and noise reduction in particular, have become even more central to the research and development of efficient, adaptive and intelligent mobile communication and information processing systems. The third edition of this book has been revised extensively and improved in several ways to take account of the recent advances in theory and application of digital signal processing. The existing chapters have been updated with new materials added. Two new chapters have been introduced; one for speech enhancement in mobile noisy conditions and the other for modelling and combating noise and fading in wireless communication systems.

The aim of this book is to provide a coherent and structured presentation of the theory and applications of statistical signal processing and noise reduction methods and is organised in 17 chapters.

Chapter 1 begins with an introduction to signal processing, and provides a brief review of signal processing methodologies and applications. The basic operations of sampling and quantisation are reviewed in this chapter.

Chapter 2 provides an introduction to noise and distortion. Several different types of noise, including thermal noise, shot noise, acoustic noise, electromagnetic noise and channel distortions, are considered. The chapter concludes with an introduction to the modelling of noise processes.

Chapter 3 provides an introduction to the theory and applications of probability models and stochastic signal processing. The chapter begins with an introduction to random signals, stochastic processes, probabilistic models and statistical measures. The concepts of stationary,

nonstationary and ergodic processes are introduced in this chapter, and some important classes of random processes, such as Gaussian, mixture Gaussian, Markov chains and Poisson processes, are considered. The effects of transformation of a signal on its statistical distribution are considered.

Chapter 4 is on Bayesian estimation and classification. In this chapter the estimation problem is formulated within the general framework of Bayesian inference. The chapter includes Bayesian theory, classical estimators, the estimate–maximise method, the Cramer–Rao bound on the minimum–variance estimate, Bayesian classification, and the modelling of the space of a random signal. This chapter provides a number of examples on Bayesian estimation of signals observed in noise.

Chapter 5 considers hidden Markov models (HMMs) for nonstationary signals. The chapter begins with an introduction to the modelling of nonstationary signals and then concentrates on the theory and applications of hidden Markov models. The hidden Markov model is introduced as a Bayesian model, and methods of training HMMs and using them for decoding and classification are considered. The chapter also includes the application of HMMs in noise reduction.

Chapter 6 considers Wiener filters. The least square error filter is formulated first through minimisation of the expectation of the squared error function over the space of the error signal. Then a block-signal formulation of Wiener filters and a vector space interpretation of Wiener filters are considered. The frequency response of the Wiener filter is derived through minimisation of mean square error in the frequency domain. Some applications of the Wiener filter are considered, and a case study of the Wiener filter for removal of additive noise provides useful insight into the operation of the filter.

Chapter 7 considers adaptive filters. The chapter begins with the state-space equation for Kalman filters. The optimal filter coefficients are derived using the principle of orthogonality of the innovation signal. The recursive least square (RLS) filter, which is an exact sample-adaptive implementation of the Wiener filter, is derived in this chapter. Then the steepest-descent search method for the optimal filter is introduced. The chapter concludes with a study of the LMS adaptive filters.

Chapter 8 considers linear prediction and sub-band linear prediction models. Forward prediction, backward prediction and lattice predictors are studied. This chapter introduces a modified predictor for the modelling of the short-term and the pitch period correlation structures. A maximum *a posteriori* (MAP) estimate of a predictor model that includes the prior probability density function of the predictor is introduced. This chapter concludes with the application of linear prediction in signal restoration.

Chapter 9 considers frequency analysis and power spectrum estimation. The chapter begins with an introduction to the Fourier transform, and the role of the power spectrum in identification of patterns and structures in a signal process. The chapter considers nonparametric spectral estimation, model-based spectral estimation, the maximum entropy method, and high-resolution spectral estimation based on eigenanalysis.

Chapter 10 considers interpolation of a sequence of unknown samples. This chapter begins with a study of the ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples, and the factors that affect interpolation. Interpolators are divided into two categories: polynomial and statistical interpolators. A general form of polynomial interpolation as well as its special forms (Lagrange, Newton, Hermite and cubic spline interpolators) is considered. Statistical interpolators in this chapter include maximum

a posteriori interpolation, least square error interpolation based on an autoregressive model, time–frequency interpolation, and interpolation through the search of an adaptive codebook for the best signal.

Chapter 11 considers spectral subtraction. A general form of spectral subtraction is formulated and the processing distortions that result from spectral subtraction are considered. The effects of processing distortions on the distribution of a signal are illustrated. The chapter considers methods for removal of the distortions and also nonlinear methods of spectral subtraction. This chapter concludes with an implementation of spectral subtraction for signal restoration.

Chapters 12 and 13 cover the modelling, detection and removal of impulsive noise and transient noise pulses. In Chapter 12, impulsive noise is modelled as a binary-state nonstationary process and several stochastic models for impulsive noise are considered. For removal of impulsive noise, median filters and a method based on a linear prediction model of the signal process are considered. The materials in Chapter 13 closely follow Chapter 12. In Chapter 13, a template-based method, an HMM-based method and an AR model-based method for removal of transient noise are considered.

Chapter 14 covers echo cancellation. The chapter begins with an introduction to telephone line echoes, and considers line echo suppression and adaptive line echo cancellation. Then the problem of acoustic echoes and acoustic coupling between loudspeaker and microphone systems is considered. The chapter concludes with a study of a sub-band echo cancellation system.

Chapter 15 covers blind deconvolution and channel equalisation. This chapter begins with an introduction to channel distortion models and the ideal channel equaliser. Then the Wiener equaliser, blind equalisation using the channel input power spectrum, blind deconvolution based on linear predictive models, Bayesian channel equalisation and blind equalisation for digital communication channels are considered. The chapter concludes with equalisation of maximum phase channels using higher-order statistics.

Chapter 16 covers speech enhancement methods. Speech enhancement in noisy environments improves the quality and intelligibility of speech for human communication and increases the accuracy of automatic speech recognition systems. Noise reduction systems are increasingly important in a range of applications such as mobile phones, hands-free phones, teleconferencing systems and in-car cabin communication systems. This chapter provides an overview of the main methods for single-input and multiple-input speech enhancement in noise.

Chapter 17 covers the issue of noise in wireless communication. Noise, fading and limited radio bandwidth are the main factors that constrain the capacity and the speed of communication on wireless channels. Research and development of communications systems aim to increase the spectral efficiency, defined as the data bits per second per Hertz bandwidth of a communication channel. For improved efficiency, modern mobile communications systems rely on signal processing methods at almost every stage from source coding to the allocation of time bandwidth and space resources. In this chapter we consider how communications signal processing methods are employed for improving the speed and capacity of communications systems.

As an additional resource, this book is supported by a companion website on which lecturers and instructors can find electronic versions of the figures. Please go to <ftp://ftp.wiley.co.uk/pub/books/vaseghi3e>.

Symbols

A	Matrix of predictor coefficients
a_k	Linear predictor coefficients
\mathbf{a}	Linear predictor coefficients vector
a_{ij}	Probability of transition from state i to state j in a Markov model
$\alpha_i(t)$	Forward probability in an HMM
$b(m)$	Backward prediction error
$b(m)$	Binary state signal
$\beta_i(t)$	Backward probability in an HMM
$c_{xx}(m)$	Covariance of signal $x(m)$
$c_{XX}(k_1, k_2, \dots, k_N)$	k th-order cumulant of $x(m)$
$C_{XX}(\omega_1, \omega_2, \dots, \omega_{K-1})$	k th-order cumulant spectra of $x(m)$
D	Diagonal matrix
$e(m)$	Estimation error
$E[x]$	Expectation of x
f	Frequency variable
F_s	Sampling frequency
$f_X(\mathbf{x})$	Probability density function for process X
$f_{X,Y}(\mathbf{x}, \mathbf{y})$	Joint probability density function of X and Y
$f_{X Y}(\mathbf{x} \mathbf{y})$	Probability density function of X conditioned on Y
$f_{X;\boldsymbol{\theta}}(x; \boldsymbol{\theta})$	Probability density function of X with $\boldsymbol{\theta}$ as a parameter
$f_{X S,\mathcal{M}}(\mathbf{x} s, \mathcal{M})$	Probability density function of X given a state sequence s of an HMM \mathcal{M} of the process X
$\Phi(m, m-1)$	State transition matrix in Kalman filter
G	Filter gain factor
\mathbf{h}	Filter coefficient vector, channel response
\mathbf{h}_{\max}	Maximum-phase channel response
\mathbf{h}_{\min}	Minimum-phase channel response
\mathbf{h}^{inv}	Inverse channel response
$H(f)$	Channel frequency response
$H^{\text{inv}}(f)$	Inverse channel frequency response

\mathbf{H}	Observation matrix, distortion matrix
\mathbf{I}	Identity matrix
\mathbf{J}	Fisher's information matrix
$ \mathbf{J} $	Jacobian of a transformation
$\mathbf{K}(m)$	Kalman gain matrix
λ	Eigenvalue
Λ	Diagonal matrix of eigenvalues
m	Discrete time index
m_k	k th-order moment
\mathcal{M}	A model, e.g. an HMM
μ	Adaptation convergence factor
μ_x	Expected mean of vector \mathbf{x}
$n(m)$	Noise
$\mathbf{n}(m)$	A noise vector of N samples
$n_i(m)$	Impulsive noise
$N(f)$	Noise spectrum
$N^*(f)$	Complex conjugate of $N(f)$
$\overline{N(f)}$	Time-averaged noise spectrum
$\mathcal{N}(\mathbf{x}, \mu_{xx}, \Sigma_{xx})$	A Gaussian pdf with mean vector μ_{xx} and covariance matrix Σ_{xx}
$O(\cdot)$	In the order of (\cdot)
P	Filter order (length)
$P_X(x_i)$	Probability mass function of x_i
$P_{X,Y}(x_i, y_j)$	Joint probability mass function of x_i and y_j
$P_{X Y}(x_i y_j)$	Conditional probability mass function of x_i given y_j
$P_{NN}(f)$	Power spectrum of noise $n(m)$
$P_{XX}(f)$	Power spectrum of the signal $x(m)$
$P_{XY}(f)$	Cross-power spectrum of signals $x(m)$ and $y(m)$
θ	Parameter vector
$\hat{\theta}$	Estimate of the parameter vector θ
r_k	Reflection coefficients
$r_{xx}(m)$	Autocorrelation function
$\mathbf{r}_{xx}(m)$	Autocorrelation vector
\mathbf{R}_{xx}	Autocorrelation matrix of signal $\mathbf{x}(m)$
\mathbf{R}_{xy}	Cross-correlation matrix
s	State sequence
s^{ML}	Maximum-likelihood state sequence
σ_n^2	Variance of noise $n(m)$
Σ_{nn}	Covariance matrix of noise $\mathbf{n}(m)$
Σ_{xx}	Covariance matrix of signal $\mathbf{x}(m)$
σ_x^2	Variance of signal $x(m)$
σ_n^2	Variance of noise $n(m)$
$x(m)$	Clean signal
$\hat{x}(m)$	Estimate of clean signal
$\mathbf{x}(m)$	Clean signal vector
$X(f)$	Frequency spectrum of signal $x(m)$
$X^*(f)$	Complex conjugate of $X(f)$

$\overline{X(f)}$	Time-averaged frequency spectrum of the signal $x(m)$
$X(f, t)$	Time-frequency spectrum of the signal $x(m)$
\mathbf{X}	Clean signal matrix
\mathbf{X}^H	Hermitian transpose of \mathbf{X}
$y(m)$	Noisy signal
$\mathbf{y}(m)$	Noisy signal vector
$\hat{\mathbf{y}}(m m-i)$	Prediction of $\mathbf{y}(m)$ based on observations up to time $m-i$
\mathbf{Y}	Noisy signal matrix
\mathbf{Y}^H	Hermitian transpose of \mathbf{Y}
Var	Variance
w_k	Wiener filter coefficients
$\mathbf{w}(m)$	Wiener filter coefficients vector
$W(f)$	Wiener filter frequency response
z	z -transform variable

Abbreviations

AR	Autoregressive process
ARMA	Autoregressive moving average process
AWGN	Additive white Gaussian noise
bps	Bits per second
cdf	Cumulative density function
CELP	Code excited linear prediction
dB	Decibels: $10 \log_{10}(\text{power ratio})$
DFT	Discrete Fourier transform
DSP	Digital signal processing
EM	Estimate–maximise
ESPIRIT	Estimation of signal parameters via rotational invariance techniques
FFT	Fast Fourier transform
FIR	Finite impulse response
GMM	Gaussian mixture model
GSM	Global system for mobile communications
HMM	Hidden Markov model
Hz	Hertz, unit of frequency in cycles per second
IFFT	Inverse fast Fourier transform
IID	Independent identically distributed
IIR	Infinite impulse response
ISD	Itakura–Saito distance
ISI	Inter symbol interference
LMS	Least mean squared error
LP	Linear prediction model
LPSS	Spectral subtraction based on linear prediction model
LS	Least square
LSAR	Least square AR interpolation
LSE	Least square error
LTI	Linear time invariant
MA	Moving average process
MAP	Maximum <i>a posteriori</i> estimate

<i>M</i> -ary	Multilevel signalling
MAVE	Minimum absolute value of error estimate
MIMO	Multiple-input multiple-output
ML	Maximum likelihood estimate
MMSE	Minimum mean squared error estimate
ms	Milliseconds
MUSIC	Multiple signal classification
NLMS	Normalised least mean squared error
pdf	Probability density function
pmf	Probability mass function
psd	Power spectral density
QRD	Orthogonal matrix decomposition
RF	Radio frequency
RLS	Recursive least square
SINR	Signal-to-impulsive noise ratio
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
SVD	Singular value decomposition
Var	Variance

1



Introduction

1.1 Signals and Information	1
1.2 Signal Processing Methods	3
1.3 Applications of Digital Signal Processing	5
1.4 Sampling and Analogue-to-digital Conversion	17
Bibliography	21

Signal processing provides the basic analysis, modelling and synthesis tools for a diverse area of technological fields, including telecommunication, artificial intelligence, biological computation and system identification. Signal processing is concerned with the modelling, detection, identification and utilisation of patterns and structures in a signal process. Applications of signal processing methods include audio hi-fi, digital TV and radio, cellular mobile phones, voice recognition, vision, radar, sonar, geophysical exploration, medical electronics, bio-signal processing and in general any system that is concerned with the communication or processing and retrieval of information. Signal processing theory plays a central role in the development of digital telecommunication and automation systems, and in the efficient transmission, reception and decoding of information.

This chapter begins with a definition of signals, and a brief introduction to various signal processing methodologies. We consider several key applications of digital signal processing in adaptive noise reduction, channel equalisation, pattern classification/recognition, audio signal coding, signal detection, spatial processing for directional reception of signals, Dolby noise reduction and radar.

1.1 SIGNALS AND INFORMATION

A signal is the variation of a quantity by which information is conveyed regarding the state, the characteristics, the composition, the trajectory, the evolution, the course of action or the

intention of the information source. A signal is a means of conveying information regarding the state(s) of a variable.

The information conveyed in a signal may be used by humans or machines for communication, forecasting, decision-making, control, geophysical exploration, medical diagnosis, forensics, etc. The types of signals that signal processing deals with include textual data, audio, ultrasonic, subsonic, image, electromagnetic, medical, biological, financial and seismic signals.

Figure 1.1 illustrates a communication system composed of an information source, $I(t)$, followed by a system, $T[\cdot]$, for transformation of the information into variation of a signal, $x(t)$, a communication channel, $h[\cdot]$, for propagation of the signal from the transmitter to the receiver, additive channel noise, $n(t)$, and a signal processing unit at the receiver for extraction of the information from the received signal.

In general, there is a mapping operation that maps the output, $I(t)$, of an information source to the signal, $x(t)$, that carries the information; this mapping operator may be denoted as $T[\cdot]$ and expressed as

$$x(t) = T[I(t)] \quad (1.1)$$

The information source $I(t)$ is normally discrete-valued, whereas the signal $x(t)$ that carries the information to a receiver may be continuous or discrete. For example, in multimedia communication the information from a computer, or any other digital communication device, is in the form of a sequence of binary numbers (ones and zeros), which would need to be transformed into voltage or current variations and modulated to the appropriate form for transmission in a communication channel over a physical link.

As a further example, in human speech communication the voice-generating mechanism provides a means for the speaker to map each discrete word into a distinct pattern of modulation of the acoustic vibrations of air that can propagate to the listener. To communicate a word, w , the speaker generates an acoustic signal realisation of the word, $x(t)$; this acoustic signal may be contaminated by ambient noise and/or distorted by a communication channel, or impaired by the speaking abnormalities of the talker, and received as the noisy, distorted and/or incomplete signal $y(t)$, modelled as

$$y(t) = h[x(t)] + n(t) \quad (1.2)$$

In addition to conveying the spoken word, the acoustic speech signal has the capacity to convey information on the prosody (i.e. pitch, intonation and stress patterns in pronunciation) of speech and the speaking characteristics, accent and emotional state of the talker. The listener extracts this information by processing the signal $y(t)$.

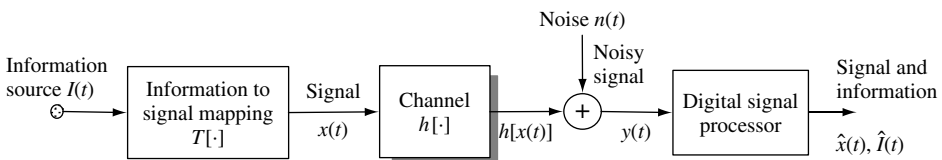


Figure 1.1 Illustration of a communication and signal processing system.

In the past few decades, the theory and applications of digital signal processing have evolved to play a central role in the development of modern telecommunication and information technology systems.

Signal processing methods are central to efficient communication, and to the development of intelligent man-machine interfaces in areas such as speech and visual pattern recognition for multimedia systems. In general, digital signal processing is concerned with two broad areas of information theory:

- (1) efficient and reliable coding, transmission, reception, storage and representation of signals in communication systems; and
- (2) extraction of information from noisy signals for pattern recognition, detection, forecasting, decision-making, signal enhancement, control, automation, etc.

In the next section we consider four broad approaches to signal processing.

1.2 SIGNAL PROCESSING METHODS

Signal processing methods have evolved in algorithmic complexity, aiming for optimal utilisation of the information in order to achieve the best performance. In general the computational requirement of signal processing methods increases, often exponentially, with the algorithmic complexity. However, the implementation cost of advanced signal processing methods has been offset and made affordable by the consistent trend in recent years of a continuing increase in the performance, coupled with a simultaneous decrease in the cost, of signal processing hardware.

Depending on the method used, digital signal processing algorithms can be categorised into one or a combination of four broad categories. These are transform-based signal processing, model-based signal processing, Bayesian statistical signal processing and neural networks, as illustrated in Figure 1.2. These methods are briefly described below.

1.2.1 TRANSFORM-BASED SIGNAL PROCESSING

The purpose of a transform is to describe a signal or a system in terms of a combination of a set of elementary simple signals (such as sinusoidal signals) that lend themselves to

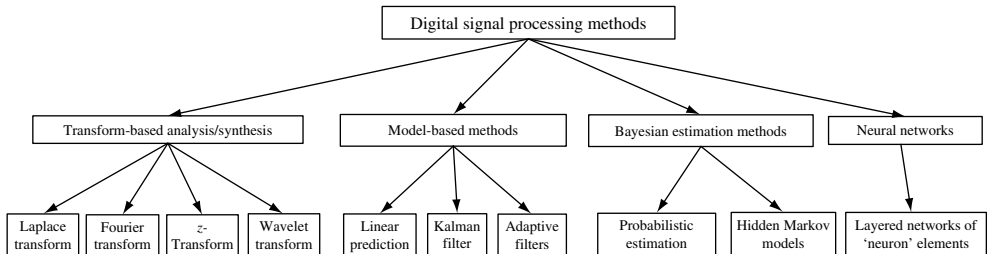


Figure 1.2 A broad categorisation of some of the most commonly used signal processing methods.

relatively easy analysis, interpretation and manipulation. Transform-based signal processing methods include Fourier transform, Laplace transform, z -transform and wavelet transforms. The most widely applied signal transform is the Fourier transform, which is effectively a form of vibration analysis, in that a signal is expressed in terms of a combination of the sinusoidal vibrations that make up the signal. Fourier transform is employed in a wide range of applications, including popular music coders, noise reduction and feature extraction for pattern recognition. The Laplace transform, and its discrete-time version the z -transform, are generalisations of the Fourier transform and describe a signal or a system in terms of a set of sinusoids with exponential amplitude envelopes.

In Fourier, Laplace and z -transform, the different sinusoidal basis functions of the transforms all have the same duration and differ in terms of their frequency of vibrations and amplitude envelopes. In contrast, the wavelets are multi-resolution transforms in which a signal is described in terms of a combination of elementary waves of different durations. The set of basis functions in a wavelet is composed of contractions and dilations of a single elementary wave. This allows non-stationary events of various durations in a signal to be identified and analysed.

1.2.2 MODEL-BASED SIGNAL PROCESSING

Model-based signal processing methods utilise a parametric model of the signal generation process. The parametric model normally describes the predictable structures and the expected patterns in the signal process, and can be used to forecast the future values of a signal from its past trajectory. Model-based methods normally outperform nonparametric methods, since they utilise more information in the form of a model of the signal process. However, they can be sensitive to the deviations of a signal from the class of signals characterised by the model. The most widely used parametric model is the linear prediction model, described in Chapter 8. Linear prediction models have facilitated the development of advanced signal processing methods for a wide range of applications such as low-bit-rate speech coding in cellular mobile telephony, digital video coding, high-resolution spectral analysis, radar signal processing and speech recognition.

1.2.3 BAYESIAN SIGNAL PROCESSING

The fluctuations of a purely random signal, or the distribution of a class of random signals in the signal space, cannot be modelled by a predictive equation, but can be described in terms of the statistical average values, and modelled by a probability distribution function in a multidimensional signal space. For example, as described in Chapter 10, a linear prediction model driven by a random signal can provide a source-filter model of the acoustic realisation of a spoken word. However, the random input signal of the linear prediction model, or the variations in the characteristics of different acoustic realisations of the same word across the speaking population, can only be described in statistical terms and in terms of probability functions.

The Bayesian inference theory provides a generalised framework for statistical processing of random signals, and for formulating and solving estimation and decision-making problems. Chapter 4 describes the Bayesian inference methodology and the estimation of random processes observed in noise.

1.2.4 NEURAL NETWORKS

Neural networks are combinations of relatively simple nonlinear adaptive processing units, arranged to have a structural resemblance to the transmission and processing of signals in biological neurons. In a neural network several layers of parallel processing elements are interconnected by a hierarchically structured connection network. The connection weights are trained to perform a signal processing function such as prediction or classification. Neural networks are particularly useful in nonlinear partitioning of a signal space, in feature extraction and pattern recognition and in decision-making systems. In some hybrid pattern recognition systems neural networks are used to complement Bayesian inference methods. Since the main objective of this book is to provide a coherent presentation of the theory and applications of statistical signal processing, neural networks are not discussed in this book

1.3 APPLICATIONS OF DIGITAL SIGNAL PROCESSING

In recent years, the development and commercial availability of increasingly powerful and affordable digital computers has been accompanied by the development of advanced digital signal processing algorithms for a wide variety of applications such as noise reduction, telecommunications, radar, sonar, video and audio signal processing, pattern recognition, geophysics explorations, data forecasting, and the processing of large databases for the identification, extraction and organisation of unknown underlying structures and patterns. Figure 1.3 shows a broad categorisation of some digital signal processing (DSP) applications. This section provides a review of several key applications of DSP methods.

1.3.1 ADAPTIVE NOISE CANCELLATION

In speech communication from a noisy acoustic environment such as a moving car or train, or over a noisy telephone channel, the speech signal is observed in an additive random noise.

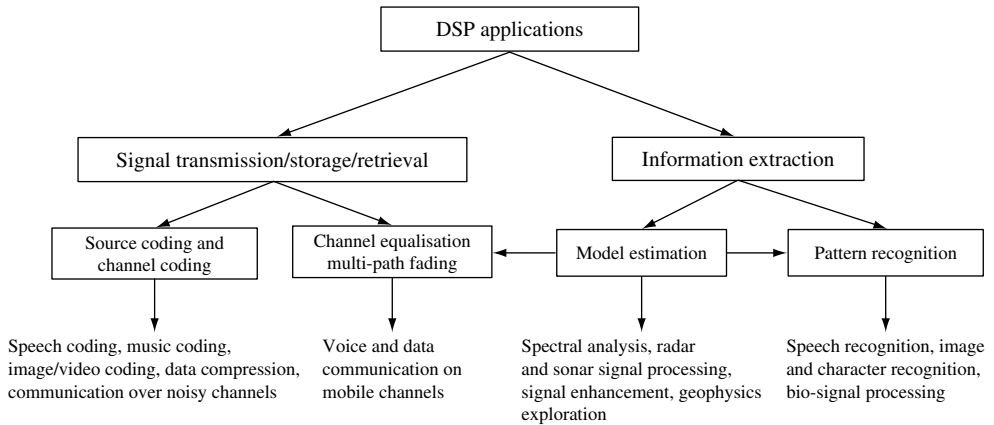


Figure 1.3 A classification of the applications of digital signal processing.

In signal measurement systems the information-bearing signal is often contaminated by noise from its surrounding environment. The noisy observation, $y(m)$, can be modelled as

$$y(m) = x(m) + n(m) \quad (1.3)$$

where $x(m)$ and $n(m)$ are the signal and the noise, and m is the discrete-time index. In some situations, for example when using a mobile telephone in a moving car, or when using a radio communication device in an aircraft cockpit, it may be possible to measure and estimate the instantaneous amplitude of the ambient noise using a directional microphone. The signal, $x(m)$, may then be recovered by subtraction of an estimate of the noise from the noisy signal.

Figure 1.4 shows a two-input adaptive noise cancellation system for enhancement of noisy speech. In this system a directional microphone takes as input the noisy signal $x(m) + n(m)$, and a second directional microphone, positioned some distance away, measures the noise $\alpha n(m + \tau)$. The attenuation factor, α , and the time delay, τ , provide a rather over-simplified model of the effects of propagation of the noise to different positions in the space where the microphones are placed. The noise from the second microphone is processed by an adaptive digital filter to make it equal to the noise contaminating the speech signal, and then subtracted from the noisy signal to cancel out the noise. The adaptive noise canceller is more effective in cancelling out the low-frequency part of the noise, but generally suffers from the nonstationary character of the signals, and from the over-simplified assumption that a linear filter can model the diffusion and propagation of the noise sound in the space.

1.3.2 ADAPTIVE NOISE REDUCTION

In many applications, for example at the receiver of a telecommunication system, there is no access to the instantaneous value of the contaminating noise, and only the noisy signal is available. In such cases the noise cannot be cancelled out, but it may be reduced, in an

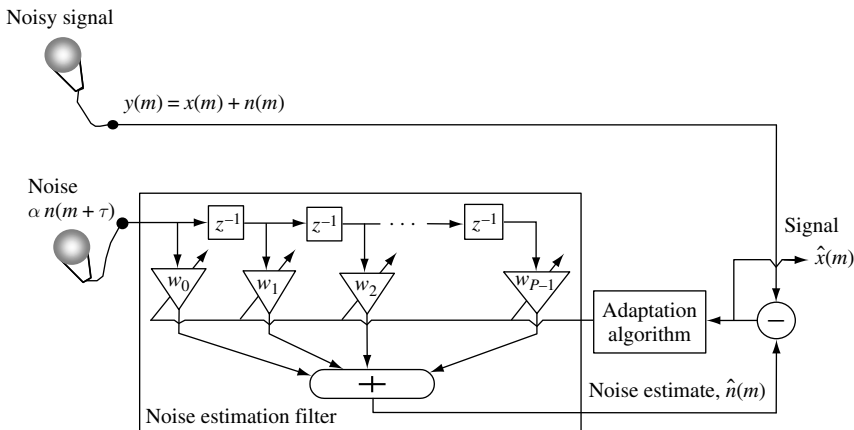


Figure 1.4 Configuration of a two-microphone adaptive noise canceller.

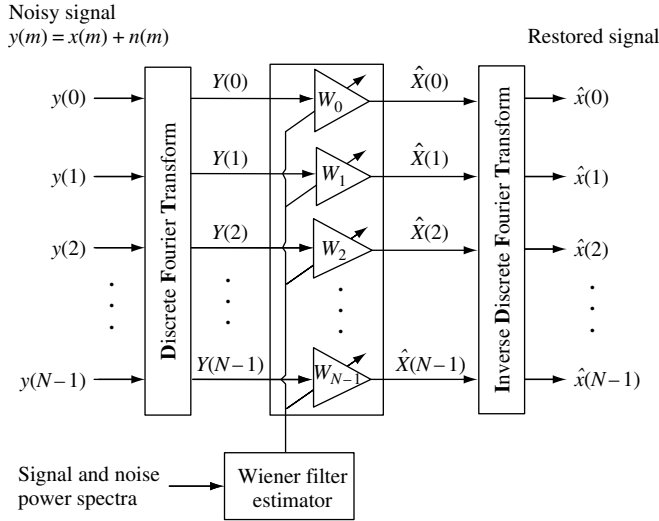


Figure 1.5 A frequency-domain Wiener filter for reducing additive noise.

average sense, using the statistics of the signal and the noise process. Figure 1.5 shows a bank of Wiener filters for reducing additive noise when only the noisy signal is available. The filter bank coefficients attenuate each noisy signal frequency in inverse proportion to the signal-to-noise ratio at that frequency. The Wiener filter bank coefficients, derived in Chapter 6, are calculated from estimates of the power spectra of the signal and the noise processes.

1.3.3 BLIND CHANNEL EQUALISATION

Channel equalisation is the recovery of a signal distorted in transmission through a communication channel with a nonflat magnitude or a nonlinear phase response. When the channel response is unknown, the process of signal recovery is called ‘blind equalisation’. Blind equalisation has a wide range of applications, for example in digital telecommunications for removal of inter-symbol interference due to nonideal channel and multipath propagation, in speech recognition for removal of the effects of the microphones and communication channels, in correction of distorted images, in analysis of seismic data and in de-reverberation of acoustic gramophone recordings.

In practice, blind equalisation is feasible only if some useful statistics of the channel input are available. The success of a blind equalisation method depends on how much is known about the characteristics of the input signal and how useful this knowledge can be in the channel identification and equalisation process. Figure 1.6 illustrates the configuration of a decision-directed equaliser. This blind channel equaliser is composed of two distinct sections: an adaptive equaliser that removes a large part of the channel distortion, followed by a nonlinear decision device for an improved estimate of the channel input. The output of the decision device is the final estimate of the channel input, and it is used as the desired

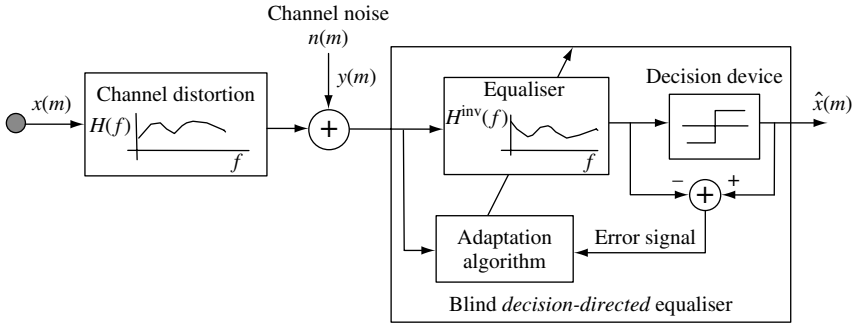


Figure 1.6 Configuration of a decision-directed blind channel equaliser.

signal to direct the equaliser adaptation process. Blind equalisation is covered in detail in Chapter 15.

1.3.4 SIGNAL CLASSIFICATION AND PATTERN RECOGNITION

Signal classification is used in detection, pattern recognition and decision-making systems. For example, a simple binary-state classifier can act as the detector of the presence, or the absence, of a known waveform in noise. In signal classification, the aim is to design a minimum-error system for labelling a signal with one of a number of likely classes of signal.

To design a classifier, a set of models is trained for the classes of signals that are of interest in the application. The simplest form that the models can assume is a bank, or code book, of waveforms, each representing the prototype for one class of signals. A more complete model for each class of signals takes the form of a probability distribution function. In the classification phase, a signal is labelled with the nearest or the most likely class. For example, in communication of a binary bit stream over a band-pass channel, the binary phase-shift keying (BPSK) scheme signals the bit '1' using the waveform $A_c \sin \omega_c t$ and the bit '0' using $-A_c \sin \omega_c t$.

At the receiver, the decoder has the task of classifying and labelling the received noisy signal as a '1' or a '0'. Figure 1.7 illustrates a correlation receiver for a BPSK signalling

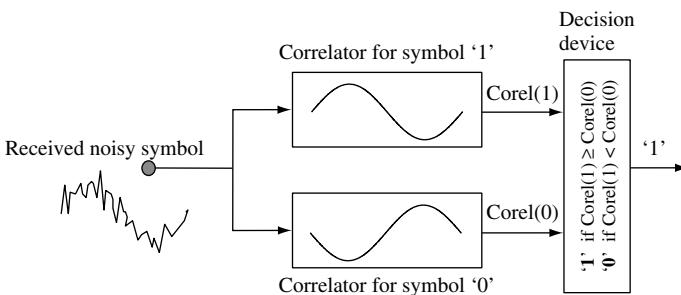


Figure 1.7 A block diagram illustration of the classifier in a binary phase-shift keying demodulation.

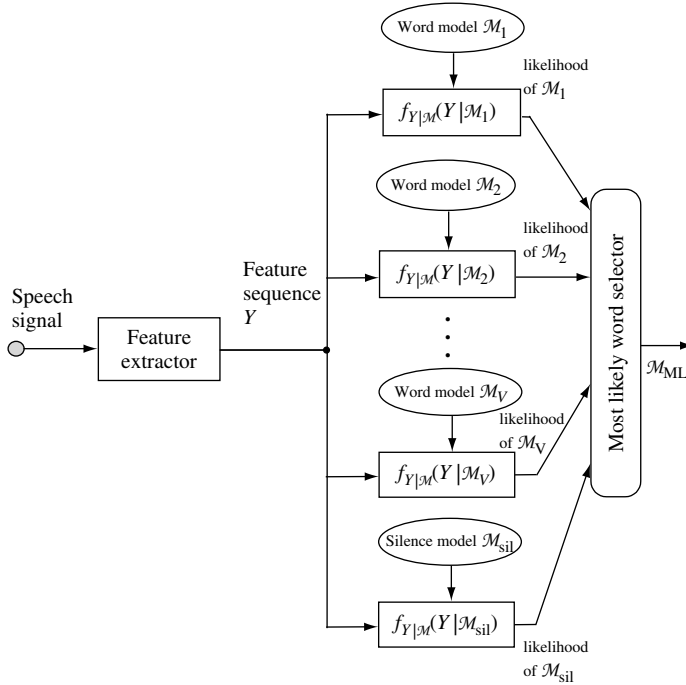


Figure 1.8 Configuration of a speech recognition system; $f(Y|M_i)$ is the likelihood of the model M_i given an observation sequence Y .

scheme. The receiver has two correlators, each programmed with one of the two symbols representing the binary states for the bit ‘1’ and the bit ‘0’. The decoder correlates the unlabelled input signal with each of the two candidate symbols and selects the candidate that has a higher correlation with the input.

Figure 1.8 illustrates the use of a classifier in a limited-vocabulary, isolated-word speech recognition system. Assume there are V words in the vocabulary. For each word a model is trained, on many different examples of the spoken word, to capture the average characteristics and the statistical variations of the word. The classifier has access to a bank of $V + 1$ models, one for each word in the vocabulary and an additional model for the silence periods. In the speech-recognition phase, the task is to decode and label an acoustic speech feature sequence, representing an unlabelled spoken word, as one of the V likely words or silence. For each candidate word the classifier calculates a probability score and selects the word with the highest score.

1.3.5 LINEAR PREDICTION MODELLING OF SPEECH

Linear predictive models are widely used in speech processing applications such as low-bit-rate speech coding in cellular telephony, speech enhancement and speech recognition. Speech is generated by inhaling air into the lungs, and then exhaling it through the vibrating

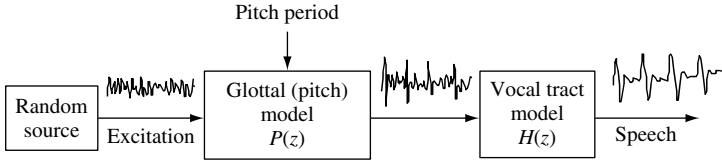


Figure 1.9 Linear predictive model of speech.

glottis cords and the vocal tract. The random, noise-like, air flow from the lungs is spectrally shaped and amplified by the vibrations of the glottal cords and the resonance of the vocal tract. The effect of the vibrations of the glottal cords and the vocal tract is to introduce a measure of correlation and predictability to the random variations of the air from the lungs. Figure 1.9 illustrates a source-filter model for speech production. The source models the lung and emits a random excitation signal which is filtered, first by a pitch filter model of the glottal cords and then by a model of the vocal tract.

The main source of correlation in speech is the vocal tract modelled by a linear predictor. A linear predictor forecasts the amplitude of the signal at time m , $x(m)$, using a linear combination of P previous samples $[x(m-1), \dots, x(m-P)]$ as

$$\hat{x}(m) = \sum_{k=1}^P a_k x(m-k) \quad (1.4)$$

where $\hat{x}(m)$ is the prediction of the signal $x(m)$, and the vector $\mathbf{a}^T = [a_1, \dots, a_P]$ is the coefficients vector of a predictor of order P . The prediction error $e(m)$, i.e. the difference between the actual sample, $x(m)$, and its predicted value, $\hat{x}(m)$, is defined as

$$e(m) = x(m) - \sum_{k=1}^P a_k x(m-k) \quad (1.5)$$

The prediction error $e(m)$ may also be interpreted as the random excitation or the so-called innovation content of $x(m)$. From Equation (1.5) a signal generated by a linear predictor can be synthesised as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \quad (1.6)$$

1.3.6 DIGITAL CODING OF AUDIO SIGNALS

In digital audio, the memory required to record a signal, the bandwidth required for signal transmission and the signal-to-quantisation noise ratio are all directly proportional to the number of bits per sample. The objective in the design of a coder is to achieve high fidelity with as few bits per sample as possible, at an affordable implementation cost. Audio signal coding schemes utilise the statistical structures of the signal and a model of the signal generation, together with information on the psychoacoustics and the masking effects of hearing. In general, there are two main categories of audio coders: model-based coders, used

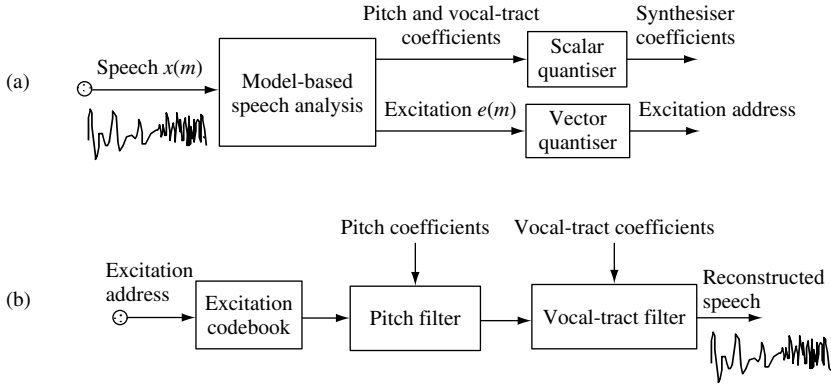


Figure 1.10 Block diagram configuration of a model-based speech (a) coder and (b) decoder.

for low-bit-rate speech coding in applications such as cellular telephony, and transform-based coders used in high-quality coding of speech and digital hi-fi audio.

Figure 1.10 shows a simplified block diagram configuration of a speech coder-decoder of the type used in digital cellular telephones. The speech signal is modelled as the output of a filter excited by a random signal. The random excitation models the air exhaled through the lung, and the filter models the vibrations of the glottal cords and the vocal tract. At the transmitter, speech is segmented into blocks about 30 ms long, during which speech parameters can be assumed to be stationary. Each block of speech samples is analysed to extract and transmit a set of excitation and filter parameters that can be used to synthesise the speech. At the receiver, the model parameters and the excitation are used to reconstruct the speech.

A transform-based coder is shown in Figure 1.11. The aim of transformation is to convert the signal into a form that lends itself to more convenient and useful interpretation and manipulation. In Figure 1.11 the input signal is transformed to the frequency domain using

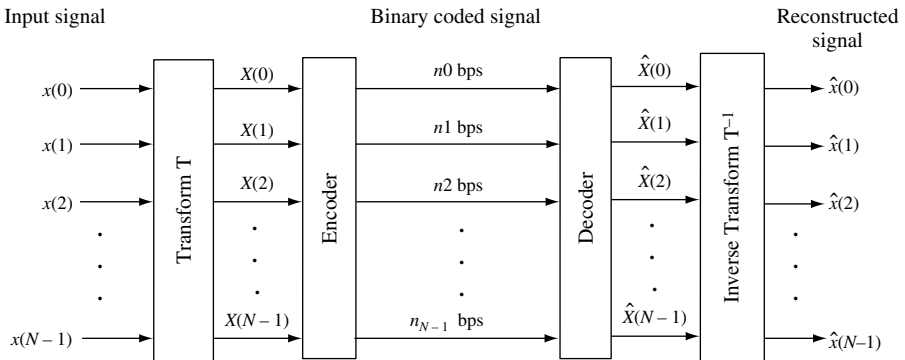


Figure 1.11 Illustration of a transform-based coder.

a filter bank, or a discrete Fourier transform, or a discrete cosine transform. The three main advantages of coding a signal in the frequency domain are:

- (1) The frequency spectrum of a signal has a relatively well-defined structure, for example most of the signal power is usually concentrated in the lower regions of the spectrum.
- (2) A relatively low-amplitude frequency would be masked in the near vicinity of a large-amplitude frequency and can therefore be coarsely encoded without any audible degradation.
- (3) The frequency samples are orthogonal and can be coded independently with different precisions.

The number of bits assigned to each frequency of a signal is a variable that reflects the contribution of that frequency to the reproduction of a perceptually high-quality signal. In an adaptive coder, the allocation of bits to different frequencies is made to vary with the time variations of the power spectrum of the signal.

1.3.7 DETECTION OF SIGNALS IN NOISE

In the detection of signals in noise, the aim is to determine if the observation consists of noise alone, or if it contains a signal. The noisy observation, $y(m)$, can be modelled as

$$y(m) = b(m)x(m) + n(m) \quad (1.7)$$

where $x(m)$ is the signal to be detected, $n(m)$ is the noise and $b(m)$ is a binary-valued state indicator sequence such that $b(m) = 1$ indicates the presence of the signal, $x(m)$, and $b(m) = 0$ indicates that the signal is absent. If the signal, $x(m)$, has a known shape, then a correlator or a matched filter can be used to detect the signal, as shown in Figure 1.12. The impulse response $h(m)$ of the matched filter for detection of a signal, $x(m)$, is the time-reversed version of $x(m)$ given by

$$h(m) = x(N - 1 - m) \quad 0 \leq m \leq N - 1 \quad (1.8)$$

where N is the length of $x(m)$. The output of the matched filter is given by

$$z(m) = \sum_{k=0}^{N-1} h(m-k)y(k) \quad (1.9)$$

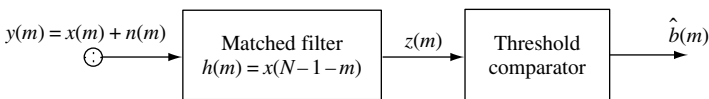


Figure 1.12 Configuration of a matched filter followed by a threshold comparator for detection of signals in noise.

Table 1.1 Four possible outcomes in a signal detection problem.

$\hat{b}(m)$	$b(m)$	Detector decision
0	0	Signal absent <i>Correct</i>
0	1	Signal absent (<i>Missed</i>)
1	0	Signal present (<i>False alarm</i>)
1	1	Signal present <i>Correct</i>

The matched filter output is compared with a threshold and a binary decision is made as

$$\hat{b}(m) = \begin{cases} 1 & \text{if } z(m) \geq \text{Threshold} \\ 0 & \text{otherwise} \end{cases} \quad (1.10)$$

where $\hat{b}(m)$ is an estimate of the binary state indicator sequence $b(m)$, and may be erroneous, particularly if the signal-to-noise ratio is low. Table 1.1 lists four possible outcomes that, together, $b(m)$ and its estimate, $\hat{b}(m)$, can assume. The choice of the threshold level affects the sensitivity of the detector. The higher the threshold, the lower the likelihood that noise would be classified as signal is, so the false alarm rate falls, but the probability of misclassification of signal as noise increases. The risk in choosing a threshold value θ can be expressed as

$$\mathcal{R}(\text{Threshold} = \theta) = P_{\text{False Alarm}}(\theta) + P_{\text{Miss}}(\theta) \quad (1.11)$$

The choice of the threshold reflects a trade-off between the misclassification rate $P_{\text{Miss}}(\theta)$ and the false alarm rate $P_{\text{False Alarm}}(\theta)$.

1.3.8 DIRECTIONAL RECEPTION OF WAVES: BEAM-FORMING

Beam-forming is the spatial processing of plane waves received by an array of sensors such that the waves' incidents at a particular spatial angle are passed through, whereas those arriving from other directions are attenuated. Beam-forming is used in radar and sonar signal processing (Figure 1.13) to steer the reception of signals towards a desired direction, and in speech processing to reduce the effects of ambient noise.

To explain the process of beam-forming, consider a uniform linear array of sensors as illustrated in Figure 1.14. The term *linear array* implies that the array of sensors is spatially arranged in a straight line and with equal spacing, d , between the sensors. Consider a sinusoidal far-field plane wave with a frequency F_0 propagating towards the sensors at an incidence angle of θ , as illustrated in Figure 1.14. The array of sensors samples the incoming wave as it propagates in space. The time delay for the wave to travel a distance of d between two adjacent sensors is given by

$$\tau = \frac{d \sin \theta}{c} \quad (1.12)$$

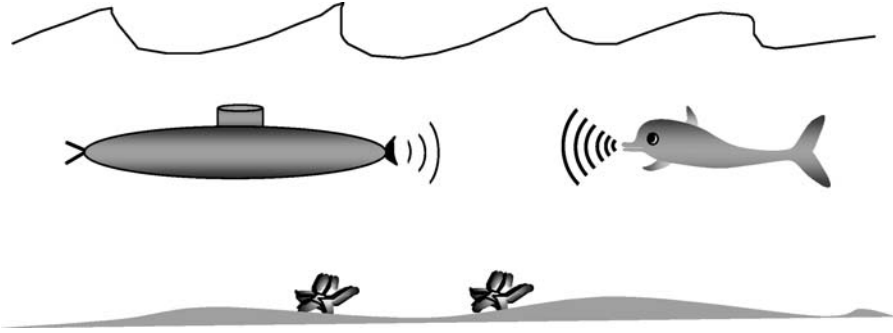


Figure 1.13 Sonar: detection of objects using the intensity and time delay of reflected sound waves.

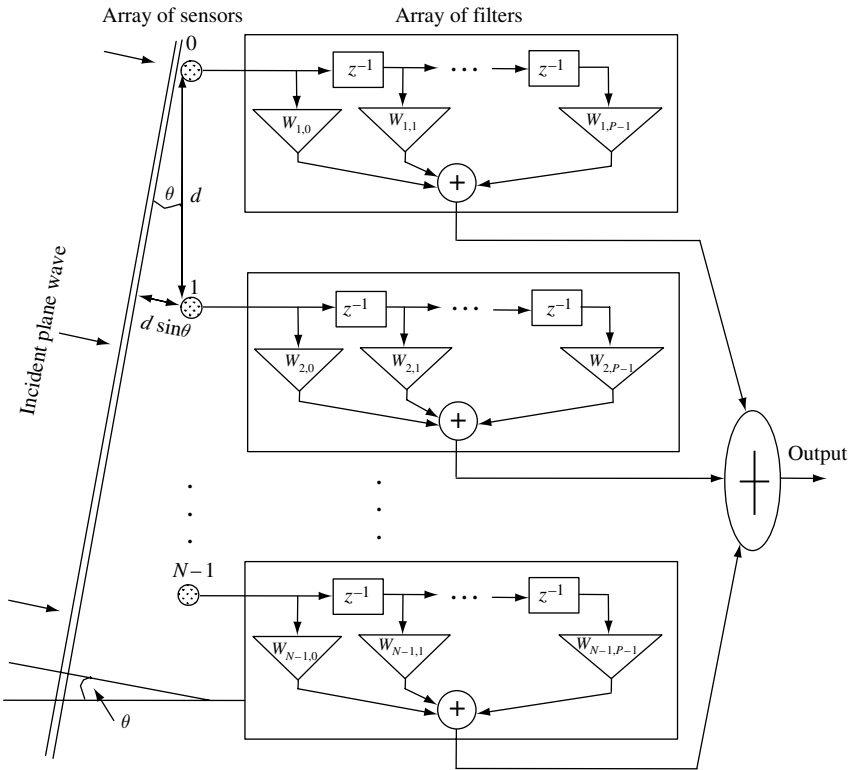


Figure 1.14 Illustration of a beam-former, for directional reception of signals.

where c is the speed of propagation of the wave in the medium. The phase difference corresponding to a delay of τ is given by

$$\varphi = 2\pi \frac{\tau}{T_0} = 2\pi F_0 \frac{d \sin \theta}{c} \tag{1.13}$$

where T_0 is the period of the sine wave. By inserting appropriate corrective time delays in the path of the samples at each sensor, and then averaging the outputs of the sensors, the signals arriving from direction θ will be time-aligned and coherently combined, whereas those arriving from other directions will suffer cancellations and attenuations. Figure 1.14 illustrates a beam-former as an array of digital filters arranged in space. The filter array acts as a two-dimensional space-time signal processing system. The space filtering allows the beam-former to be steered in a desired direction, for example in the direction along which the incoming signal has the maximum intensity. The phase of each filter controls the time delay, and can be adjusted to coherently combine the signals. The magnitude frequency response of each filter can be used to remove the out-of-band noise.

1.3.9 DOLBY NOISE REDUCTION

Dolby noise-reduction systems work by boosting the energy and the signal-to-noise ratio of the high-frequency spectrum of audio signals. The energy of audio signals is mostly concentrated in the low-frequency part of the spectrum (below 2 kHz). The higher frequencies that convey quality and sensation have relatively low energy, and can be degraded by even a small amount of noise. For example, when a signal is recorded on a magnetic tape, the tape 'hiss' noise affects the quality of the recorded signal. On playback, the higher-frequency parts of an audio signal recorded on a tape have a smaller signal-to-noise ratio than the low-frequency parts. Therefore noise at high frequencies is more audible and less masked by the signal energy. Dolby noise reduction systems broadly work on the principle of emphasising and boosting the low energy of the high-frequency signal components prior to recording the signal. When a signal is recorded, it is processed and encoded using a combination of a pre-emphasis filter and dynamic range compression. At playback, the signal is recovered using a decoder based on a combination of a de-emphasis filter and a decompression circuit. The encoder and decoder must be well matched and cancel each other out in order to avoid processing distortion.

Dolby developed a number of noise-reduction systems designated Dolby A, Dolby B and Dolby C. These differ mainly in the number of bands and the pre-emphasis strategy that they employ. Dolby A, developed for professional use, divides the signal spectrum into four frequency bands: band 1 is low-pass and covers 0 to 80 Hz; band 2 is band-pass and covers 80 Hz to 3 kHz; band 3 is high-pass and covers above 3 kHz; and band 4 is also high-pass and covers above 9 kHz. At the encoder the gain in each band is adaptively adjusted to boost low-energy signal components. Dolby A provides a maximum gain of 10–15 dB in each band if the signal level falls 45 dB below the maximum recording level. The Dolby B and Dolby C systems are designed for consumer audio systems, and use two bands instead of the four bands used in Dolby A. Dolby B provides a boost of up to 10 dB when the signal level is low (less than 45 dB below the maximum reference) and Dolby C provides a boost of up to 20 dB, as illustrated in Figure 1.15.

1.3.10 RADAR SIGNAL PROCESSING: DOPPLER FREQUENCY SHIFT

Figure 1.16 shows a simple diagram of a radar system that can be used to estimate the range and speed of an object such as a moving car or aeroplane. A radar system consists of a

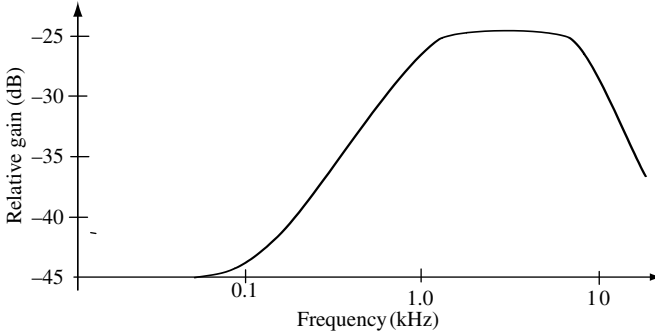


Figure 1.15 Illustration of the pre-emphasis response of Dolby C: up to 20 dB boost is provided when the signal falls 45 dB below maximum recording level.

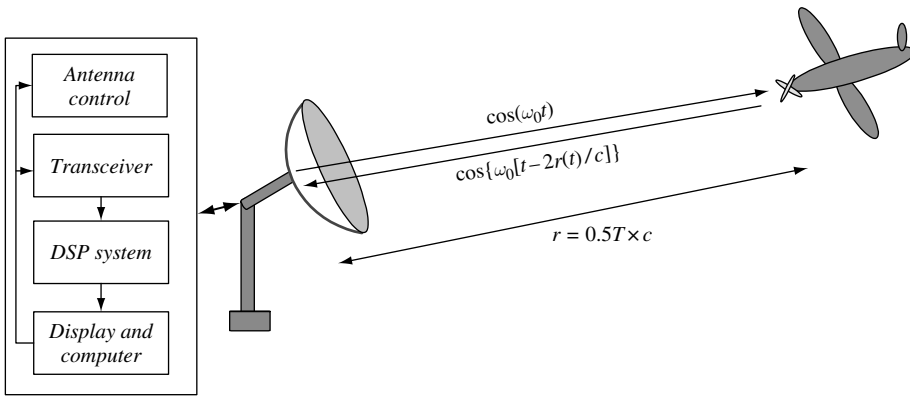


Figure 1.16 Illustration of a radar system.

transceiver (transmitter/receiver) that generates and transmits sinusoidal pulses at microwave frequencies. The signal travels with the speed of light and is reflected back from any object in its path. The analysis of the echo received provides information such as range, speed and acceleration. The received signal has the form

$$x(t) = A(t) \cos\{\omega_0[t - 2r(t)/c]\} \tag{1.14}$$

where $A(t)$, the time-varying amplitude of the reflected wave, depends on the position and characteristics of the target, $r(t)$ is the time-varying distance of the object from the radar and c is the velocity of light. The time-varying distance of the object can be expanded in a Taylor series as

$$r(t) = r_0 + \dot{r}t + \frac{1}{2!}\ddot{r}t^2 + \frac{1}{3!}\ddot{\ddot{r}}t^3 + \dots \tag{1.15}$$

where r_0 is the distance, \dot{r} the velocity, \ddot{r} the acceleration, etc. Approximating $r(t)$ with the first two terms of the Taylor series expansion, we have

$$r(t) \approx r_0 + \dot{r}t \quad (1.16)$$

Substituting Equation (1.15) in Equation (1.13) yields

$$x(t) = A(t) \cos[(\omega_0 - 2\dot{r}\omega_0/c)t - 2\omega_0 r_0/c] \quad (1.17)$$

Note that the frequency of reflected wave is shifted by an amount

$$\omega_d = 2\dot{r}\omega_0/c \quad (1.18)$$

This shift in frequency is known as the Doppler frequency. If the object is moving towards the radar then the distance $r(t)$ is decreasing with time, \dot{r} is negative, and an increase in the frequency is observed. Conversely, if the object is moving away from the radar, then the distance $r(t)$ is increasing, \dot{r} is positive, and a decrease in the frequency is observed. Thus the frequency analysis of the reflected signal can reveal information on the direction and speed of the object. The distance r_0 is given by

$$r_0 = 0.5T \times c \quad (1.19)$$

where T is the round-trip time for the signal to hit the object and arrive back at the radar and c is the velocity of light.

1.4 SAMPLING AND ANALOGUE-TO-DIGITAL CONVERSION

A digital signal is a sequence of real-valued or complex-valued numbers, representing the fluctuations of an information-bearing quantity with time, space or some other variable. The *basic* elementary discrete-time signal is the unit-sample signal, $\delta(m)$, defined as

$$\delta(m) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \quad (1.20)$$

where m is the discrete time index. A digital signal, $x(m)$, can be expressed as the sum of a number of amplitude-scaled and time-shifted unit samples as

$$x(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(m-k) \quad (1.21)$$

Figure 1.17 illustrates a discrete-time signal. Many random processes, such as speech, music, radar and sonar, generate signals that are continuous in time and continuous in amplitude. Continuous signals are termed ‘analogue’ because their fluctuations with time are analogous to the variations of the signal source. For digital processing, analogue signals are sampled, and each sample is converted into an n -bit digit. The digitisation process should be performed such that the original signal can be recovered from its digital version with no loss of

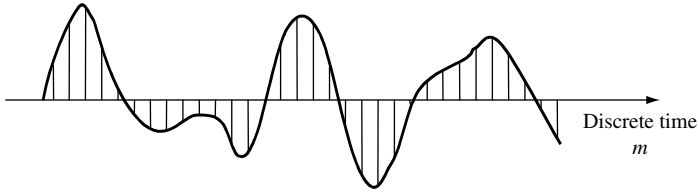


Figure 1.17 A discrete-time signal and its envelope of variation with time.

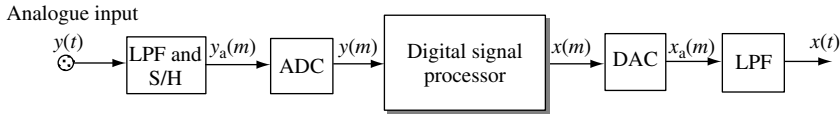


Figure 1.18 Configuration of a digital signal processing system.

information, and with as high a fidelity as is required in an application. Figure 1.18 illustrates a block diagram configuration of a digital signal processor with an analogue input. The low-pass filter (LPF) removes out-of-band signal frequencies above a pre-selected range. The sample-and-hold (S/H) unit periodically samples the signal to convert the continuous-time signal into a discrete-time signal.

The analogue-to-digital converter (ADC) maps each continuous amplitude sample into an n -bit digit. After processing, the digital output of the processor can be converted back into an analogue signal using a digital-to-analogue converter (DAC) and a low-pass filter, as illustrated in Figure 1.18.

1.4.1 SAMPLING AND RECONSTRUCTION OF ANALOGUE SIGNALS

The conversion of an analogue signal to a sequence of n -bit digits consists of two basic steps of sampling and quantisation. The sampling process, when performed with sufficiently high speed, can capture the fastest fluctuations of the signal, and can be a loss-less operation in that the analogue signal can be recovered through interpolation of the sampled sequence, as described in Chapter 10. The quantisation of each sample into an n -bit digit involves some irrevocable error and possible loss of information. However, in practice the quantisation error can be made negligible by using an appropriately high number of bits as in a digital audio hi-fi. A sampled signal can be modelled as the product of a continuous-time signal, $x(t)$, and a periodic impulse train, $p(t)$, as

$$\begin{aligned} x_{\text{sampled}}(t) &= x(t)p(t) \\ &= \sum_{m=-\infty}^{\infty} x(t) \delta(t - mT_s) \end{aligned} \quad (1.22)$$

where T_s is the sampling interval and the sampling function, $p(t)$, is defined as

$$p(t) = \sum_{m=-\infty}^{\infty} \delta(t - mT_s) \quad (1.23)$$

The spectrum, $P(f)$, of the sampling function, $p(t)$, is also a periodic impulse train given by

$$P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \quad (1.24)$$

where $F_s = 1/T_s$ is the sampling frequency. Since multiplication of two time-domain signals is equivalent to the convolution of their frequency spectra, we have

$$X_{\text{sampled}}(f) = FT[x(t) \cdot p(t)] = X(f) * P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \quad (1.25)$$

where the operator $FT[.]$ denotes the Fourier transform. In Equation (1.25) the convolution of a signal spectrum $X(f)$ with each impulse, $\delta(f - kF_s)$, shifts $X(f)$ and centres it on kF_s . Hence, as expressed in Equation (1.25), the sampling of a signal $x(t)$ results in a periodic repetition of its spectrum $X(f)$ centred on frequencies $0, \pm F_s, \pm 2F_s, \dots$

When the sampling frequency, F_s , is higher than twice the maximum frequency content of the signal, BH_2 (i.e. $F_s > 2B$), then the repetitions of the signal spectra are separated, as shown in Figure 1.19. In this case, the analogue signal can be recovered by passing the sampled signal through an analogue low-pass filter with a cut-off frequency of just above BH_2 . If the sampling frequency is less than $2B$ (i.e. $F_s < 2B$), then the adjacent repetitions of the spectrum overlap and the original spectrum cannot be recovered. The distortion, due to an insufficiently high sampling rate, is irrevocable and is known as *aliasing*.

This observation is the basis of the *Nyquist sampling theorem*, which states: a band-limited continuous-time signal, with a highest frequency content (bandwidth) of B Hz, can be recovered from its samples provided that the sampling frequency $F_s > 2B$ samples per second.

In practice, sampling is achieved using an electronic switch that allows a capacitor to charge up or down to the level of the input voltage once every T_s seconds, as illustrated in Figure 1.20. As illustrated in Figure 1.19, the staircase shape of a sample-and-hold signal can be obtained by filtering the idealised impulse-train sampled signal through a filter with a rectangular impulse response.

1.4.2 QUANTISATION

For digital signal processing, continuous-amplitude samples from the sample-and-hold are quantised and mapped into n -bit binary digits. For quantisation to n bits, the amplitude range of the signal is divided into 2^n discrete levels, and each sample is quantised to the nearest quantisation level, and then mapped to the binary code assigned to that level. Figure 1.21 illustrates the quantisation of a signal into four discrete levels. Quantisation is a many-to-one mapping, in that all the values that fall within the continuum of a quantisation band are

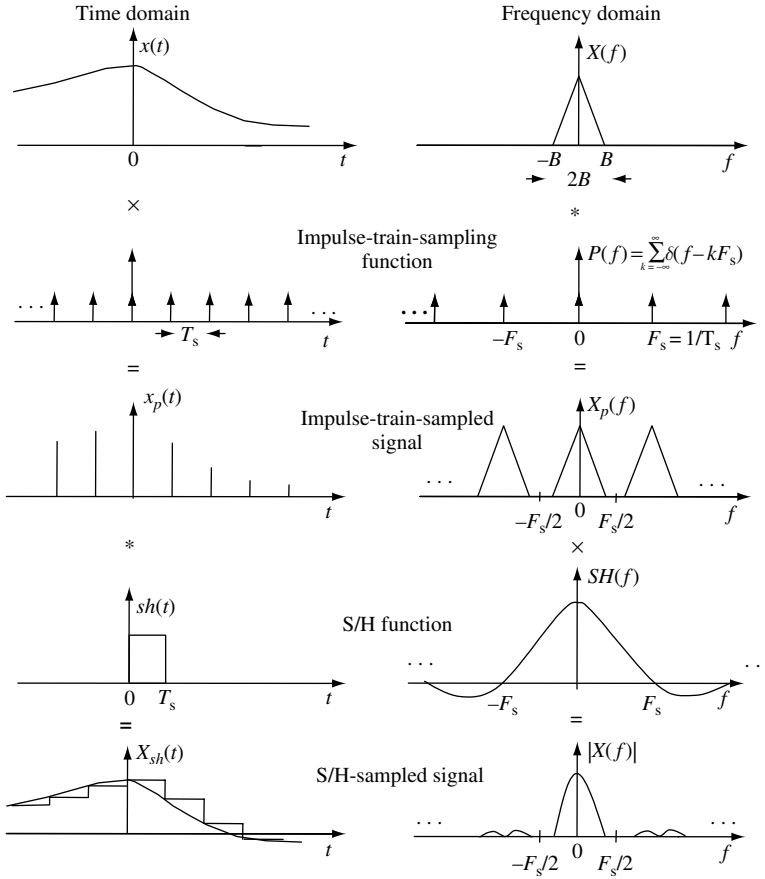


Figure 1.19 Sample-and-hold signal modelled as impulse-train sampling followed by convolution with a rectangular pulse.

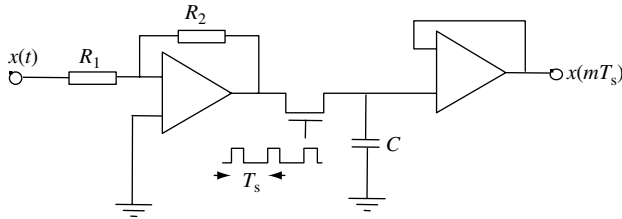


Figure 1.20 A simplified sample-and-hold circuit diagram.

mapped to the centre of the band. The mapping between an analogue sample, $x_a(m)$, and its quantised value, $x(m)$, can be expressed as

$$x(m) = Q[x_a(m)] \tag{1.26}$$

where $Q[\cdot]$ is the quantising function.

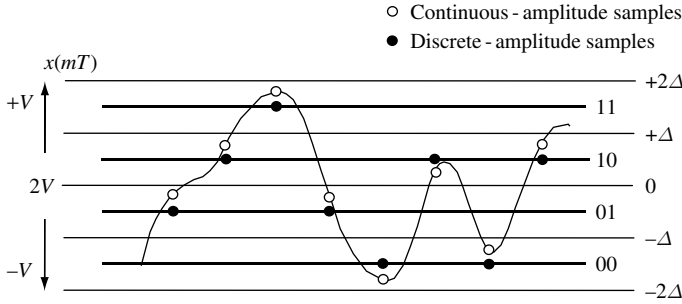


Figure 1.21 Illustration of offset-binary scalar quantisation.

The performance of a quantiser is measured by signal-to-quantisation noise ratio, SQNR, per bit. The quantisation noise is defined as

$$e(m) = x(m) - x_a(m) \quad (1.27)$$

Now consider an n -bit quantiser with an amplitude range of $\pm V$ volts. The quantisation step size is $\Delta = 2V/2^n$. Assuming that the quantisation noise is a zero-mean uniform process with an amplitude range of $\pm\Delta/2$, we can express the noise power as

$$\begin{aligned} \mathcal{E}[e^2(m)] &= \int_{-\Delta/2}^{\Delta/2} f_E[e(m)]e^2(m) de(m) = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2(m) de(m) \\ &= \frac{\Delta^2}{12} = \frac{V^2 2^{-2n}}{3} \end{aligned} \quad (1.28)$$

where $f_E[e(m)] = 1/\Delta$ is the uniform probability density function of the noise. Using Equation (1.27), the signal-to-quantisation noise ratio is given by

$$\begin{aligned} \text{SQNR}(n) &= 10 \log_{10} \left\{ \frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]} \right\} = 10 \log_{10} \left(\frac{P_{\text{Signal}}}{V^2 2^{-2n}/3} \right) \\ &= 10 \log_{10} 3 - 10 \log_{10} \left(\frac{V^2}{P_{\text{Signal}}} \right) + 10 \log_{10} 2^{2n} \\ &= 4.77 - \alpha + 6n \end{aligned} \quad (1.29)$$

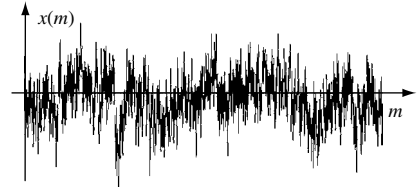
where P_{signal} is the mean signal power and α is the ratio in decibels of the peak signal power, V^2 , to the mean signal power, P_{signal} . Therefore, from Equation (1.28), every additional bit in an analogue-to-digital converter results in 6 dB improvement in signal-to-quantisation noise ratio.

BIBLIOGRAPHY

- Alexander S.T. (1986) *Adaptive Signal Processing Theory and Applications*. Springer-Verlag, New York.
 Davenport W.B. and Root W.L. (1958) *An Introduction to the Theory of Random Signals and Noise*. McGraw-Hill, New York.

- Ephraim Y. (1992) Statistical model based speech enhancement systems. *Proc. IEEE*, **80**(10): 1526–1555.
- Gallager R.G. (1968) *Information Theory and Reliable Communication*. Wiley, New York.
- Gauss K.G. (1963) *Theory of Motion of Heavenly Bodies*. Dover, New York.
- Haykin S. (1985) *Array Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.
- Haykin S. (1991) *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Kailath T. (1980) *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ.
- Kalman R.E. (1960) A new approach to linear filtering and prediction problems. *Trans. ASME Ser. D J. Basic Engng*, **82**: 35–45.
- Kay S.M. (1993) *Fundamentals of Statistical Signal Processing, Estimation Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Lim J.S. (1983) *Speech Enhancement*. Prentice Hall, Englewood Cliffs, NJ.
- Lucky R.W., Salz J. and Weldon E.J. (1968) *Principles of Data Communications*. McGraw-Hill, New York.
- Kung S.Y. (1993) *Digital Neural Networks*. Prentice-Hall, Englewood Cliffs, NJ.
- Marple S.L. (1987) *Digital Spectral Analysis with Applications*. Prentice-Hall, Englewood Cliffs, NJ.
- Oppenheim A.V. and Schaffer R.W. (1989) *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.
- Proakis J.G., Rader C.M., Ling F. and Nikias C.L. (1992) *Advanced Signal Processing*. Macmillan, New York.
- Rabiner L.R. and Gold B. (1975) *Theory and Applications of Digital Processing*. Prentice-Hall, Englewood Cliffs, NJ.
- Rabiner L.R. and Schaffer R.W. (1978) *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ.
- Scharf L.L. (1991) *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison Wesley, Reading, MA.
- Therrien C.W. (1992) *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.
- Van-Trees H.L. (1971) *Detection, Estimation and Modulation Theory. Parts I, II and III*. Wiley, New York.
- Shannon C.E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**: 379–423, 623–656.
- Widrow B. (1975) Adaptive noise cancelling: principles and applications. *Proc. IEEE*, **63**: 1692–1716.
- Wiener N. (1948) *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. MIT Press, Cambridge, MA.
- Wiener N. (1949) *Cybernetics*. MIT Press, Cambridge, MA.
- Wilsky A.S. (1979) *Digital Signal Processing, Control and Estimation Theory: Points of Tangency, Areas of Intersection and Parallel Directions*. MIT Press, Cambridge, MA.

2



Noise and Distortion

2.1 Introduction	24
2.2 White Noise	25
2.3 Coloured Noise	26
2.4 Impulsive Noise	27
2.5 Transient Noise Pulses	29
2.6 Thermal Noise	30
2.7 Shot Noise	31
2.8 Electromagnetic Noise	31
2.9 Channel Distortions	32
2.10 Echo and Multipath Reflections	33
2.11 Modelling Noise	33
Bibliography	37

Noise can be defined as an unwanted signal that interferes with the communication or measurement of another signal. A noise itself is a signal that conveys information regarding the source of the noise. For example, the noise from a car engine conveys information regarding the state of the engine and how smoothly it is running. The sources of noise are many and varied and include thermal noise intrinsic to electric conductors, shot noise inherent in electric current flows, audio-frequency acoustic noise emanating from moving, vibrating or colliding sources such as revolving machines, moving vehicles, computer fans, keyboard clicks, wind, rain, etc. and radio-frequency electromagnetic noise that can interfere with the transmission and reception of voice, image and data over the radio-frequency spectrum. Signal distortion is the term often used to describe a systematic undesirable change in a signal and refers to changes in a signal due to the nonideal characteristics of the communication channel, reverberations, echo, multipath reflections and missing samples.

Noise and distortion are the main factors limiting the capacity of data transmission in telecommunications and accuracy in signal measurement systems. Therefore the modelling and removal of the effects of noise and distortions have been at the core of the theory and

practice of communications and signal processing. Noise reduction and distortion removal are important problems in applications such as cellular mobile communications, speech recognition, image processing, medical signal processing, radar and sonar, and in any application where the signals cannot be isolated from noise and distortion. In this chapter, we study the characteristics and modelling of several different forms of noise.

2.1 INTRODUCTION

Noise may be defined as any unwanted signal that interferes with the communication, measurement, perception or processing of an information-bearing signal. Noise is present in various degrees in almost all environments. For example, in a digital cellular mobile telephone system, there may be several varieties of noise that could degrade the quality of communication, such as acoustic background noise, thermal noise, shot noise, electromagnetic radio-frequency noise, co-channel radio interference, radio-channel distortion, acoustic and line echoes, multipath reflection, fading and signal processing noise.

Noise can cause transmission errors and may even disrupt a communication process; hence noise processing is an important and integral part of modern telecommunications and signal processing systems. The success of a noise processing method depends on its ability to characterise and model the noise process, and to use the noise characteristics advantageously to differentiate the signal from the noise.

Depending on its source, a noise can be classified into a number of categories, indicating the broad physical nature of the noise, as follows:

- (1) Acoustic noise – emanates from moving, vibrating or colliding sources and is the most familiar type of noise present to various degrees in everyday environments. Acoustic noise is generated by such sources as moving cars, air-conditioners, computer fans, traffic, people talking in the background, wind, rain, etc.
- (2) Thermal noise and shot noise – thermal noise is generated by the random movements of thermally energised particles in an electric conductor. Thermal noise is intrinsic to all conductors and is present without any applied voltage. Shot noise consists of random fluctuations of the electric current in an electrical conductor and is intrinsic to current flow. Shot noise is caused by the fact that the current is carried by discrete charges (i.e. electrons) with random fluctuations and random arrival times.
- (3) Electromagnetic noise – present at all frequencies and in particular at the radio frequency range (kHz to GHz range) where telecommunications take place. All electric devices, such as radio and television transmitters and receivers, generate electromagnetic noise.
- (4) Electrostatic noise – generated by the presence of a voltage with or without current flow. Fluorescent lighting is one of the more common sources of electrostatic noise.
- (5) Channel distortions, echo and fading – due to nonideal characteristics of communication channels. Radio channels, such as those at GHz frequencies used by cellular mobile phone operators, are particularly sensitive to the propagation characteristics of the channel environment and fading of signals.
- (6) Processing noise – the noise that results from the digital-to-analogue processing of signals, e.g. quantisation noise in digital coding of speech or image signals, or lost data packets in digital data communication systems.

Depending on its frequency spectrum or time characteristics, a noise process can be further classified into one of several categories as follows:

- (1) White noise – purely random noise that has a flat power spectrum. White noise theoretically contains all frequencies in equal intensity.
- (2) Band-limited white noise – a noise with a flat spectrum and a limited bandwidth that usually covers the limited spectrum of the device or the signal of interest.
- (3) Narrowband noise – a noise process with a narrow bandwidth such as a 50–60 Hz ‘hum’ from the electricity supply.
- (4) Coloured noise – nonwhite noise or any wideband noise whose spectrum has a nonflat shape; examples are pink noise, brown noise and autoregressive noise.
- (5) Impulsive noise – consists of short-duration pulses of random amplitude and random duration.
- (6) Transient noise pulses – consists of relatively long duration noise pulses.

2.2 WHITE NOISE

White noise is defined as an uncorrelated random noise process with equal power at all frequencies (Figure 2.1). A random noise that has the same power at all frequencies in the range of $\pm\infty$ would necessarily need to have infinite power, and is therefore only a theoretical concept. However a band-limited noise process, with a flat spectrum covering the frequency range of a band-limited communication system, is to all intents and purposes from the point of view of the system a white noise process. For example, for an audio system with a bandwidth of 10 kHz, any flat-spectrum audio noise with a bandwidth of equal to or greater than 10 kHz looks like white noise.

The autocorrelation function of a continuous-time zero-mean white noise process with a variance of σ^2 is a delta function [Figure 2.1(b)] given by

$$r_{NN}(\tau) = \mathcal{E}[N(t)N(t + \tau)] = \sigma^2\delta(\tau) \tag{2.1}$$

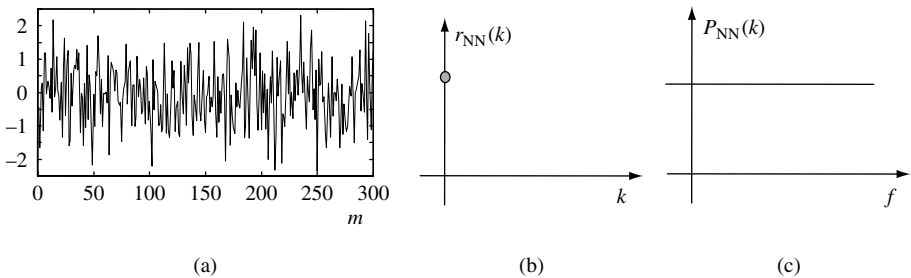


Figure 2.1 (a) Illustration of white noise. (b) Its autocorrelation function is a delta function. (c) Its power spectrum is constant.

The power spectrum of a white noise, obtained by taking the Fourier transform of Equation (2.1), is given by

$$P_{\text{NN}}(f) = \int_{-\infty}^{\infty} r_{\text{NN}}(t) e^{-j2\pi ft} dt = \sigma^2 \quad (2.2)$$

Equation (2.2) and Figure 2.1(c) show that a white noise has a constant power spectrum.

2.2.1 BAND-LIMITED WHITE NOISE

Pure white noise is a theoretical concept, since it would need to have infinite power to cover an infinite range of frequencies. Furthermore, a discrete-time signal by necessity has to be band-limited, with its highest frequency less than half the sampling rate. A more practical concept is band-limited white noise, defined as a noise with a flat spectrum in a limited bandwidth. The spectrum of band-limited white noise with a bandwidth of B Hz is given by

$$P_{\text{NN}}(f) = \begin{cases} \sigma^2, & |f| \leq B \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

Thus the total power of a band-limited white noise process is $2B\sigma^2$. The autocorrelation function of a discrete-time band-limited white noise process has the shape of a sinc function and is given by

$$r_{\text{NN}}(T_s k) = 2B\sigma^2 \frac{\sin(2\pi B T_s k)}{2\pi B T_s k} \quad (2.4)$$

where T_s is the sampling period. For convenience of notation, T_s is usually assumed to be unity. For the case when $T_s = 1/2B$, i.e. when the sampling rate is equal to the Nyquist rate, Equation (2.4) becomes

$$r_{\text{NN}}(T_s k) = 2B\sigma^2 \frac{\sin(\pi k)}{\pi k} = 2B\sigma^2 \delta(k) \quad (2.5)$$

In Equation (2.5) the autocorrelation function is a delta function.

2.3 COLOURED NOISE

Although the concept of white noise provides a reasonably realistic and mathematically convenient and useful approximation to some predominant noise processes encountered in telecommunications systems, many other noise processes are nonwhite. The term ‘coloured noise’ refers to any broadband noise with a nonwhite spectrum. For example most audio-frequency noise, such as the noise from moving cars, noise from computer fans, electric drill noise and people talking in the background, has a nonwhite predominantly low-frequency spectrum. Also, a white noise passing through a channel is ‘coloured’ by the shape of the frequency response of the channel. Two classic varieties of coloured noise are so-called ‘pink noise’ and ‘brown noise’, shown in Figures 2.2 and 2.3.

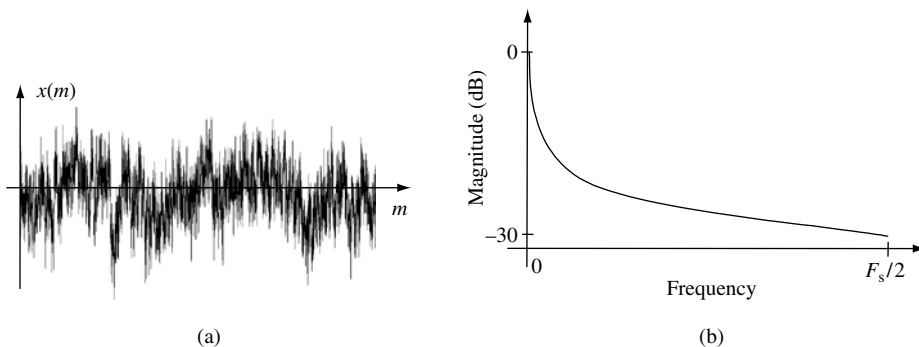


Figure 2.2 (a) A pink noise signal and (b) its magnitude spectrum.

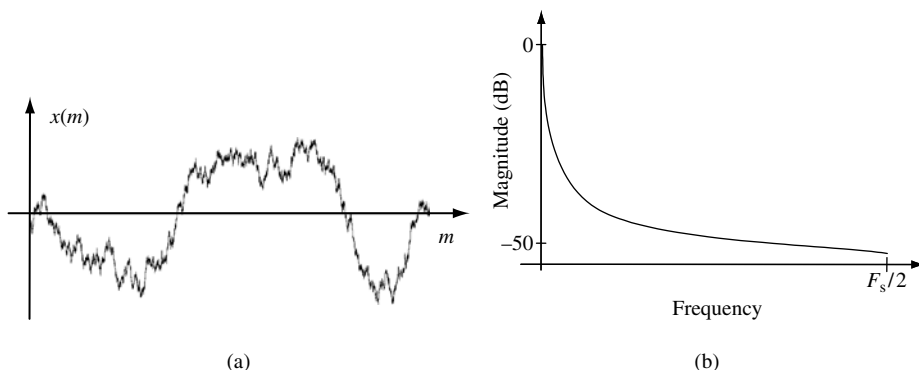


Figure 2.3 (a) A brown noise signal and (b) its magnitude spectrum.

2.4 IMPULSIVE NOISE

Impulsive noise consists of random short-duration ‘on/off’ noise pulses, caused by a variety of sources, such as switching noise, electromagnetic interference, adverse channel environment in a communication system, drop-outs or surface degradation of audio recordings, clicks from computer keyboards, etc.

Figure 2.4(a) shows an ideal impulse and its frequency spectrum. In communication systems, a real impulsive-type noise has a duration that is normally more than one sample long. For example, in the context of audio signals, short-duration, sharp pulses, of up to 3 ms (60 samples at a 20 kHz sampling rate) may be considered as impulsive noise. Figure 2.4(b) and (c) illustrates two examples of short-duration pulses and their respective spectra.

In a communications system, an impulsive noise originates at some point in time and space, and then propagates through the channel to the receiver. The received noise is time-dispersed and shaped by the channel, and can be considered as the channel impulse response. In general, the characteristics of a communication channel may be linear or nonlinear,

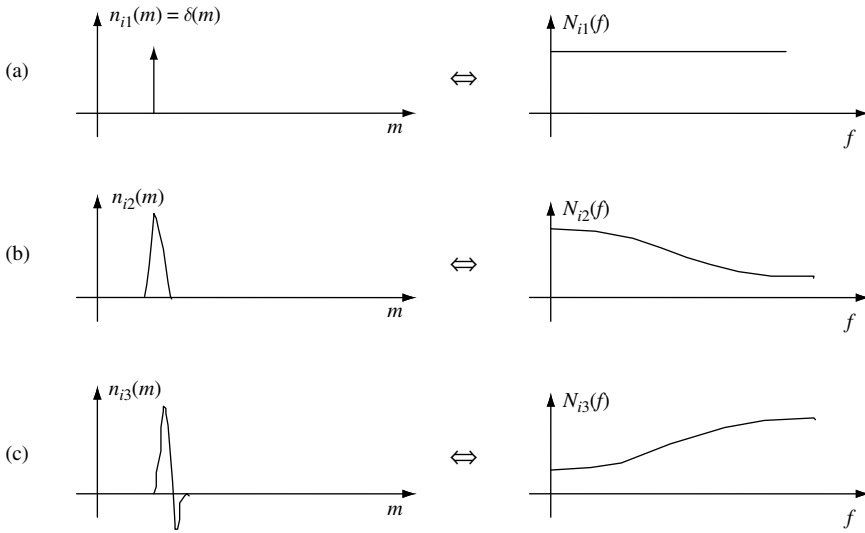


Figure 2.4 Time and frequency sketches of: (a) an ideal impulse; (b) and (c) short-duration pulses.

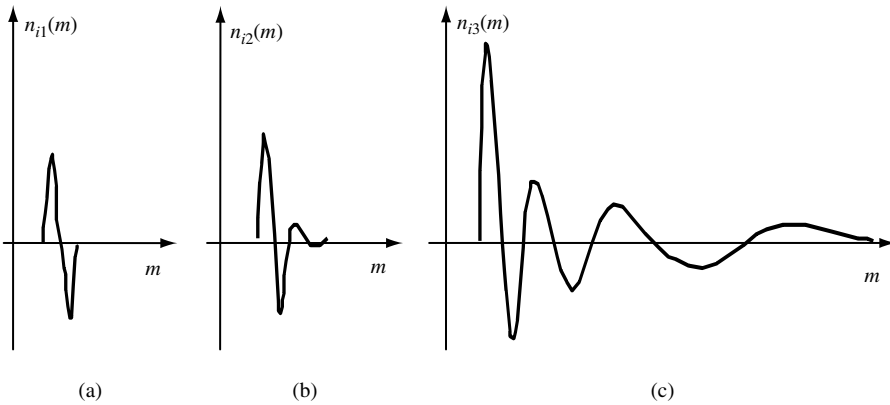


Figure 2.5 Illustration of variations of the impulse response of a nonlinear system with increasing amplitude of the impulse.

stationary or time-varying. Furthermore, many communications systems exhibit a nonlinear characteristic in response to a large-amplitude impulse.

Figure 2.5 illustrates some examples of impulsive noise, typical of that observed on an old gramophone recording. In this case, the communication channel is the playback system, and may be assumed to be time-invariant. The figure also shows some variations of the channel characteristics with the amplitude of impulsive noise. For example, in Figure 2.5(c) a large impulse excitation has generated a decaying transient pulse with time-varying period. These variations may be attributed to the nonlinear characteristics of the playback mechanism.

2.5 TRANSIENT NOISE PULSES

Transient noise pulses, observed in most communications systems, are caused by interference. Transient noise pulses often consist of a relatively short, sharp initial pulse followed by decaying low-frequency oscillations, as shown in Figure 2.6. The initial pulse is usually due to some external or internal impulsive interference, whereas the oscillations are often due to the resonance of the communication channel excited by the initial pulse, and may be considered as the response of the channel to the initial pulse. In a telecommunications system, a noise pulse originates at some point in time and space, and then propagates through the channel to the receiver. The noise pulse is shaped by the channel characteristics, and may be considered as the channel pulse response. Thus, we should be able to characterize the transient noise pulses with a similar degree of consistency as in characterizing the channels through which the pulses propagate.

As an illustration of the shape of a transient noise pulse, consider the scratch pulses from a damaged gramophone record shown in Figure 2.6(a) and (b). Scratch noise pulses are acoustic manifestations of the response of the stylus and the associated electromechanical playback system to a sharp physical discontinuity on the recording medium. Since scratches are essentially the impulse response of the playback mechanism, it is expected that, for a given system, various scratch pulses exhibit similar characteristics. As shown in Figure 2.6(b), a typical scratch pulse waveform often exhibits two distinct regions:

- (1) the initial high-amplitude pulse response of the playback system to the physical discontinuity on the record medium; followed by
- (2) decaying oscillations that cause additive distortion; the initial pulse is relatively short and has a duration on the order of 1–5 ms, whereas the oscillatory tail has a longer duration and may last up to 50 ms or more.

Note in Figure 2.6(b) that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the nonlinear modes of response of the electromechanical playback system excited by the physical scratch discontinuity. Observations of many scratch waveforms from damaged gramophone records reveals that they have a well-defined profile, and can be characterised by a relatively small number of typical templates. Scratch pulse modelling and removal is considered in detail in Chapter 13.

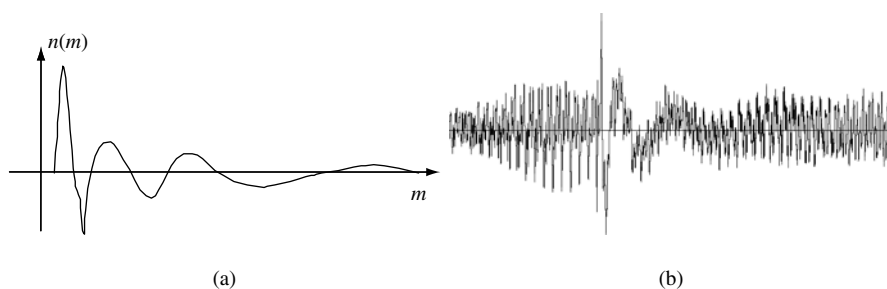


Figure 2.6 (a) A scratch pulse and music from a gramophone record. (b) The averaged profile of a gramophone record scratch pulse.

2.6 THERMAL NOISE

Thermal noise, also referred to as Johnson noise (after its discoverer, J.B. Johnson), is generated by the random movements of thermally energised (agitated) particles inside an electric conductor. Thermal noise is intrinsic to all resistors and is not a sign of poor design or manufacture, although some resistors may also have excess noise. Thermal noise cannot be circumvented by good shielding or grounding.

Note that thermal noise happens at equilibrium without the application of a voltage. The application of a voltage and the movement of current in a conductor cause an additional random fluctuation known as shot noise, as described in the next section.

The concept of thermal noise has its roots in thermodynamics and is associated with the temperature-dependent random movements of free particles such as gas molecules in a container or electrons in a conductor. Although these random particle movements average to zero, the fluctuations about the average constitute the thermal noise. For example, the random movements and collisions of gas molecules in a confined space produce random fluctuations about the average pressure. As the temperature increases, the kinetic energy of the molecules and the thermal noise increase.

Similarly, an electrical conductor contains a very large number of free electrons, together with ions that vibrate randomly about their equilibrium positions and resist the movement of the electrons. The free movement of electrons constitutes random spontaneous currents, or thermal noise, that average to zero since, in the absence of a voltage, electrons move in different directions. As the temperature of a conductor, from heat provided by its surroundings, increases, the electrons move to higher-energy states and the random current flow increases. For a metallic resistor, the mean square value of the instantaneous voltage due to the thermal noise is given by

$$\overline{v^2} = 4kTRB \quad (2.6)$$

where $k = 1.38 \times 10^{-23}$ J/k is the Boltzmann constant, T is the absolute temperature in degrees Kelvin, R is the resistance in ohms and B is the bandwidth. From Equation (2.6) and the preceding argument, a metallic resistor sitting on a table can be considered as a generator of thermal noise power, with a mean square voltage $\overline{v^2}$ and an internal resistance R . From circuit theory, the maximum available power delivered by a 'thermal noise generator', dissipated in a matched load of resistance R , is given by

$$\begin{aligned} P_N &= \overline{i^2} R = \left(\frac{v_{\text{rms}}}{2R} \right)^2 \\ R &= \frac{\overline{v^2}}{4R} = kTB \quad (\text{W}) \end{aligned} \quad (2.7)$$

where v_{rms} is the root mean square voltage. The spectral density of thermal noise is given by

$$P_N(f) = \frac{kT}{2} (\text{W/Hz}) \quad (2.8)$$

From Equation (2.8), the thermal noise spectral density has a flat shape, i.e. thermal noise is a white noise. Equation (2.8) holds well up to very high radio-frequencies of 10^{13} Hz.

2.7 SHOT NOISE

Shot noise consists of random fluctuations of the electric current in a conductor. Shot noise is caused by the fact that a current is carried by discrete electronic charges (i.e. electrons) with random arrival times. The strength of this noise increases with the increasing average current flowing through the conductor.

The term shot noise arose from the analysis of random variations in the emission of electrons from the cathode of a vacuum tube. Discrete electron particles in a current flow arrive at random times, and therefore there will be fluctuations about the average particle flow. The fluctuations in the rate of particle flow constitute the shot noise. Other instances of shot noise arise in the flow of photons in a laser beam, the flow and recombination of electrons and holes in semiconductors, and the flow of photoelectrons emitted in photodiodes.

Note that shot noise is different from thermal noise, described in Section 2.6. Thermal noise is due to ‘unforced’ random fluctuations of current (movement of particles) related to temperature and happens without any applied voltage or average current flowing. Shot noise, however, happens when there is a voltage difference and a current flow. Shot noise cannot be eliminated as it is an intrinsic part of the movement of charges that constitutes a current. In contrast, thermal noise can be reduced by reducing the operating temperature of the device.

The concept of randomness of the rate of emission or arrival of particles implies that the random variations of shot noise can be modelled by a Poisson probability distribution (see Chapter 3). The most basic statistics of shot noise, namely the mean and variance of the noise, were reported by Campbell in 1909. Rice provided an analysis of shot noise when the underlying Poisson process has a constant intensity and showed that, as the intensity of the current tends to infinity, i.e. when the average number of arrivals of charges during the observing time is large, the probability distribution of the shot noise tends to a Gaussian distribution.

Now consider an electric current as the flow of discrete electric charges. As explained, the flow of electrons is not smooth and there will be random fluctuations in the form of shot of noise. If the charges act independently of each other, it can be shown that noise current is given by

$$I_{\text{Noise}}(\text{rms}) = (2eIB)^{1/2} \quad (2.9)$$

where $e = 1.6 \times 10^{-19}$ C is the electron charge, I is the current and B is the measurement bandwidth. For example, a ‘steady’ current I of 1 A in a bandwidth 1 MHz has an rms fluctuation of $0.57 \mu\text{A}$. Equation (2.9) assumes that the charge carriers making up the current act independently. That is the case for charges crossing a barrier, such as, for example, the current in a junction diode, where the charges move by diffusion; however, it is not true for metallic conductors, where there are long-range correlations between charge carriers.

2.8 ELECTROMAGNETIC NOISE

Electromagnetic waves present in the environment constitute a level of background noise that can interfere with the operation of communication and signal processing systems. Electromagnetic waves may emanate from man-made devices or natural sources. The primary natural source of electromagnetic waves is the Sun. In the order of decreasing wavelength

and increasing frequency, various types of electromagnetic radiation include: electric motors (kHz), radio waves (kHz to GHz), microwaves (10^{11} Hz), infrared radiation (10^{13} Hz), visible light (10^{14} Hz), ultraviolet radiation (10^{15} Hz), X-rays (10^{20} Hz) and γ -radiation (10^{23} Hz).

Virtually every electrical device that generates, consumes or transmits power is a source of pollution of radio spectrum and a potential source of electromagnetic noise interference for other systems. In general, the higher the voltage or the current level, and the closer the proximity of electrical circuits/devices, the greater will be the induced noise. The common sources of electromagnetic noise are transformers, radio and television transmitters, mobile phones, microwave transmitters, a.c. power lines, motors and motor starters, generators, relays, oscillators, fluorescent lamps and electrical storms.

Electrical noise from these sources can be categorized into two basic types: electrostatic and magnetic. These two types of noise are fundamentally different, and thus require different noise-shielding measures. Unfortunately, most of the common noise sources listed above produce combinations of the two noise types, which can complicate the noise reduction problem.

Electrostatic fields are generated by the presence of voltage, with or without current flow. Fluorescent lighting is one of the more common sources of electrostatic noise. Magnetic fields are created either by the flow of electric current or by the presence of permanent magnetism. Motors and transformers are examples of the former, and the Earth's magnetic field is an instance of the latter. In order for noise voltage to be developed in a conductor, magnetic lines of flux must be cut by the conductor. Electric (and noise) generators function on this basic principle. In the presence of an alternating field, such as that surrounding a 50–60 Hz power line, voltage will be induced into any stationary conductor as the magnetic field expands and collapses. Similarly, a conductor moving through the Earth's magnetic field has a noise voltage generated in it as it cuts the lines of flux.

The main sources of electromagnetic interference in mobile communications systems are the radiations from the antennas of other mobile phones and base stations. The electromagnetic interference by mobile users and base stations can be reduced by the use of narrow-beam adaptive antennas, the so-called 'smart antennas', as described in Chapter 17.

2.9 CHANNEL DISTORTIONS

On propagating through a channel, signals are shaped, delayed and distorted by the frequency response and the attenuating (fading) characteristics of the channel. There are two main manifestations of channel distortions: magnitude distortion and phase distortion. In addition, in radio communication, we have the multipath effect, in which the transmitted signal may take several different routes to the receiver, with the effect that multiple versions of the signal with different delay and attenuation arrive at the receiver. Channel distortions can degrade or even severely disrupt a communication process, and hence channel modelling and equalization are essential components of modern digital communications systems. Channel equalization is particularly important in modern cellular communications systems, since the variations of channel characteristics and propagation attenuation in cellular radio systems are far greater than those of the landline systems.

Figure 2.7 illustrates the frequency response of a channel with one invertible and two non-invertible regions. In the noninvertible regions, the signal frequencies are heavily attenuated and lost to the channel noise. In the invertible region, the signal is distorted but recoverable.

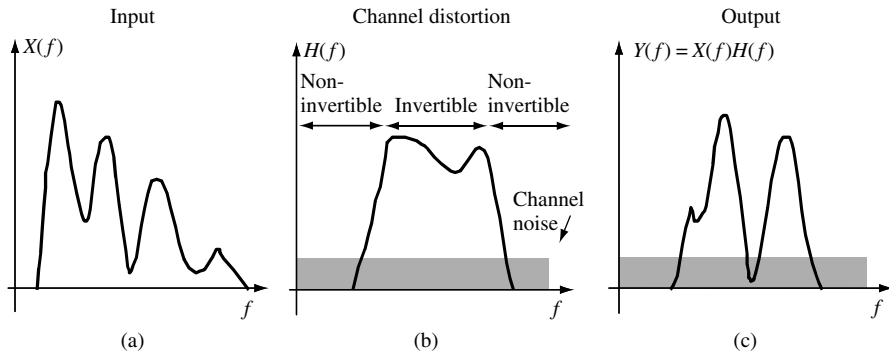


Figure 2.7 Illustration of channel distortion: (a) the input signal spectrum; (b) the channel frequency response; (c) the channel output.

This example illustrates that the channel inverse filter must be implemented with care in order to avoid undesirable results such as noise amplification at frequencies with a low signal-to-noise ratio. Channel equalization is covered in detail in Chapter 15.

2.10 ECHO AND MULTIPATH REFLECTIONS

Multipath and echo are distortions due to reflections of signals from points where the physical characteristics of the medium through which the signals propagates change. Multipath and echo happen for both acoustic and electromagnetic signals.

Echo implies that part of the signal is returned to the source. Telephone line echoes are due to the reflection of the electric signals at the point of mismatch where the two-wire subscriber line is converted to the four-wire trunk lines. Acoustic echoes are due to feedback between the speakers and microphones. Cancellation of line and acoustic echoes remain important issues in modern communications systems and are discussed in Chapter 14.

Multipath implies that the transmitted signal arrives at the destination after reflections from several different points or surfaces and through a number of different paths. In room acoustics, multipath propagation of sound waves causes reverberation of sounds. In cellular mobile communications environments, multipath propagation can cause severe distortion of the signals if it is not modelled and compensated for. Chapter 17 provides an introduction to multipath effects in mobile communications systems.

2.11 MODELLING NOISE

The objective of modelling is to characterise the structures and the patterns in a signal or a noise process. To model a noise accurately, we need a structure for modelling both the temporal and the spectral characteristics of the noise. Accurate modelling of noise statistics is the key to high-quality noisy signal classification and enhancement. Even the seemingly simple task of signal/noise classification is crucially dependent on the availability of good signal and noise models, and on the use of these models within a Bayesian framework. Hidden Markov models (described in Chapter 5) are a good structure for modelling signals or noise.

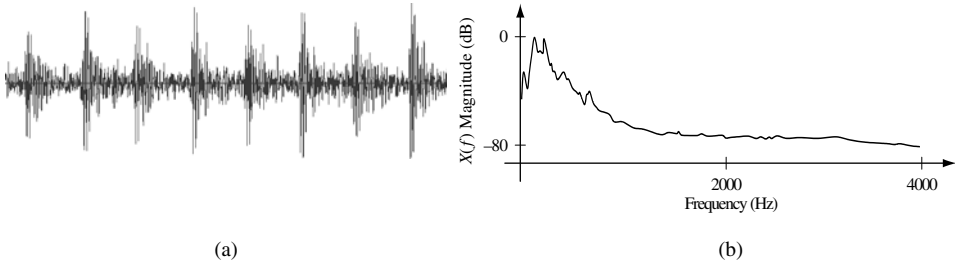


Figure 2.8 Illustration of: (a) the time-waveform of a drill noise, and (b) the frequency spectrum of the drill noise.

One of the most useful and indispensable tools for gaining insight into the structure of a noise process is the use of Fourier transform for frequency analysis. Figure 2.8 illustrates the noise from an electric drill, which, as expected, has a periodic structure. The spectrum of the drilling noise shown in Figure 2.8(a) reveals that most of the noise energy is concentrated in the lower-frequency part of the spectrum. In fact, it is true of most audio signals and noise that they have a predominantly low-frequency spectrum. However, it must be noted that the relatively lower-energy high-frequency part of audio signals plays an important part in conveying sensation and quality. Figures 2.9(a) and (b) show examples of the spectra of car noise recorded from a BMW and a Volvo. The noise in a car is nonstationary and varied, and may include as sources quasiperiodic noise from the car engine and the revolving mechanical parts of the car:

- (1) noise from the surface contact of wheels and the road surface;
- (2) noise from the air flow into the car through the air ducts, windows, sunroof, etc.;
- (3) noise from passing/overtaking vehicles.

The characteristic of car noise varies with speed, road surface conditions, weather and the environment within the car.

The simplest method for noise modelling, often used in current practice, is to estimate the noise statistics from the signal-inactive periods. In optimal Bayesian signal processing

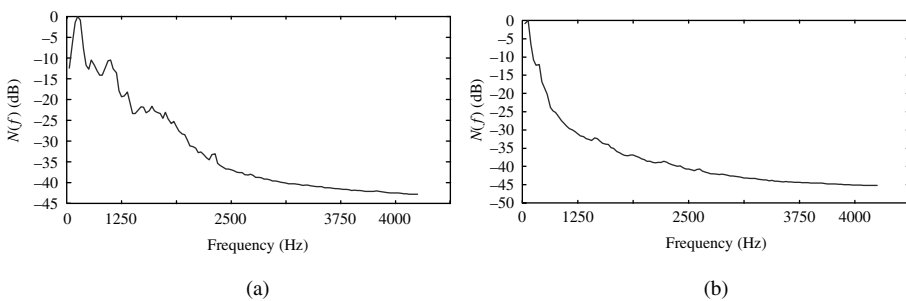


Figure 2.9 Power spectra of car noise: (a) a BMW at 70 mph and (b) a Volvo at 70 mph.

methods, a set of probability models, such as hidden Markov models (HMMs) or Gaussian mixture models (GMMs), is trained for the signal and the noise processes. The models are then used for the decoding of the underlying states of the signal and noise, and for noisy signal recognition and enhancement. Indeed, modelling noise is not, in principle, different

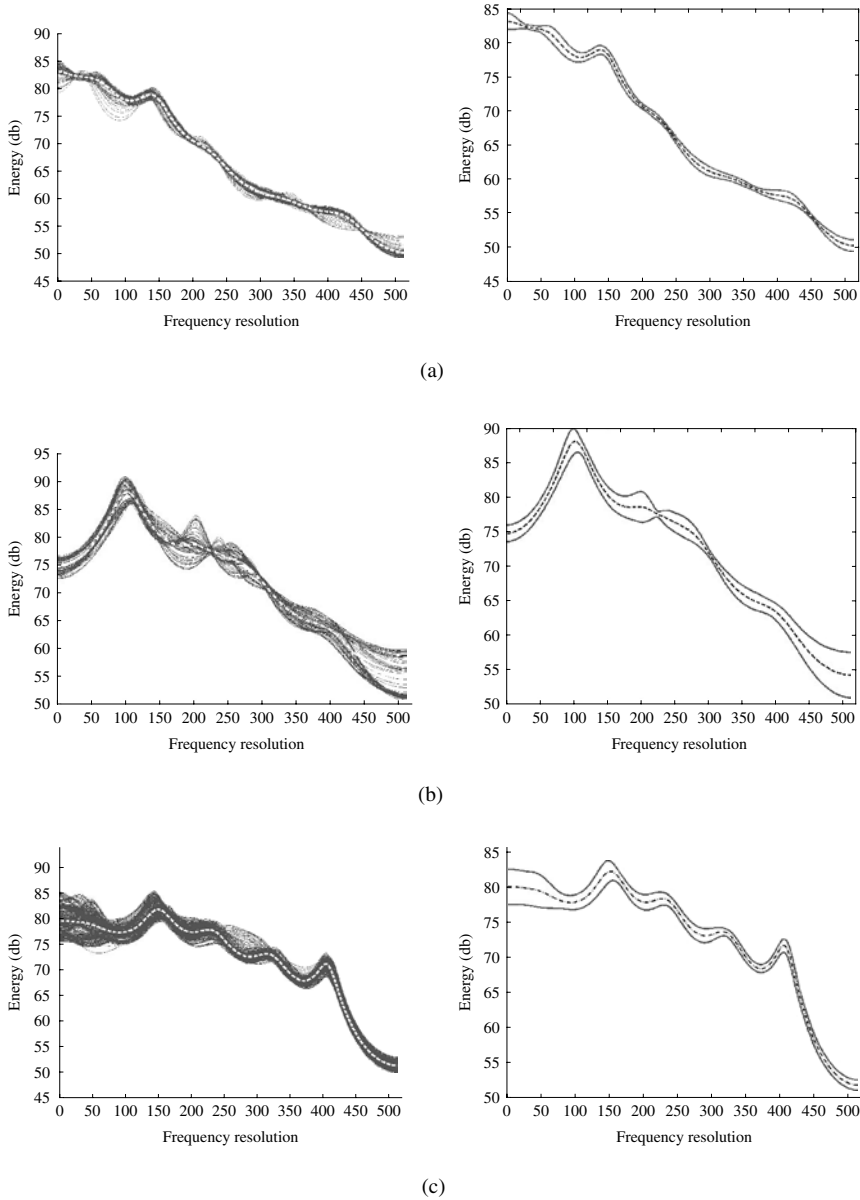


Figure 2.10 Illustration of the mean (left) and standard deviation (right) of the magnitude spectra of: (a) car noise; (b) train noise; and (c) street noise.

from modelling speech and the Bayesian inference method described in Chapter 4 and HMMs described in Chapter 5 can be applied to estimation of noise models.

Figure 2.10 illustrates the variations of the envelopes of the spectra of car noise, train noise and street noise. The spectral envelopes were obtained as the magnitude frequency responses of linear prediction models of the noise. Also shown are the mean values and the variance of the envelopes.

2.11.1 ADDITIVE WHITE GAUSSIAN NOISE MODEL

In classical communication theory, it is often assumed that the noise is a stationary additive white Gaussian (AWGN) process. Although for some problems this is a valid assumption and leads to mathematically convenient and useful solutions, in practice the noise is often time-varying, correlated and non-Gaussian. This is particularly true for impulsive-type noise and for acoustic noise, which are nonstationary and non-Gaussian and hence cannot be modelled using the AWGN assumption. Nonstationary and non-Gaussian noise processes can be modelled by a Markovian chain of stationary subprocesses as described briefly in the next section and in detail in Chapter 5.

2.11.2 HIDDEN MARKOV MODEL FOR NOISE

Most noise processes are nonstationary; that is the statistical parameters of the noise, such as its mean, variance and power spectrum, vary with time. Nonstationary processes may be modelled using the HMMs described in detail in Chapter 5. An HMM is essentially a finite-state Markov chain of stationary subprocesses. The implicit assumption in using HMMs for noise is that the noise statistics can be modelled by a Markovian chain of stationary subprocesses. Note that a stationary noise process can be modelled by a single-state HMM. For a nonstationary noise, a multistate HMM can model the time variations of the noise process with a finite number of stationary states. For non-Gaussian noise, a mixture Gaussian density model can be used to model the space of the noise within each state. In general, the number of states per model and number of mixtures per state required to accurately model a noise process depends on the nonstationary character of the noise.

An example of a nonstationary noise is the impulsive noise of Figure 2.11(a). Figure 2.11(b) shows a two-state HMM of the impulsive noise sequence: the state S_0 models the ‘impulse-off’ periods between the impulses, and state S_1 models an impulse. In those cases where

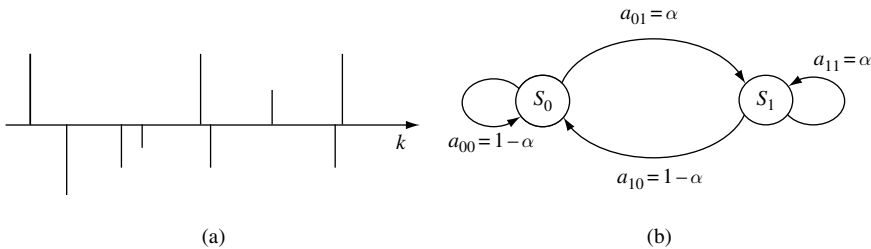


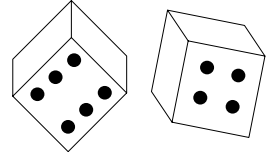
Figure 2.11 (a) An impulsive noise sequence. (b) A binary-state model of impulsive noise.

each impulse has a well-defined temporal structure, it may be beneficial to use a multi-state HMM to model the pulse itself. HMMs are used in Chapter 12 for modelling impulsive noise, and in Chapter 15 for channel equalisation.

BIBLIOGRAPHY

- Bell D.A. (1960) *Electrical Noise and Physical Mechanism*. Van Nostrand, London.
- Bennett W.R. (1960) *Electrical Noise*. McGraw-Hill, New York.
- Campbell N., (1909) The study of discontinuous phenomena. *Proc. Camb. Phil. Soc.*, **15**: 117–136.
- Davenport W.B. and Root W.L. (1958) *An Introduction to the Theory of Random Signals and Noise*. McGraw-Hill, New York.
- Ephraim Y. (1992) Statistical model based speech enhancement systems. *Proc. IEEE*, **80**(10): 1526–1555.
- Godsill S.J. (1993) The restoration of degraded audio signals. Ph.D. Thesis, Cambridge University, Cambridge.
- Rice, S. O. (1945) Mathematical analysis of random noise. *Bell Syst. Tech. J.*, **23**: 282–332, July 1944; **24**: 46–156, January 1944 [Reprinted in *Selected Papers on Noise and Stochastic Processes*, N. Wax (ed.). Dover, New York, 1954, pp. 133–294.]
- Schwartz M. (1990) *Information Transmission, Modulation and Noise*, 4th edn. McGraw-Hill, New York.
- Van-Trees H.L. (1971) *Detection, Estimation and Modulation Theory, Parts I, II and III*. Wiley, New York.

3



Probability and Information Models

3.1 Introduction	40
3.2 Random Signals	41
3.3 Probability Models	44
3.4 Information Models	50
3.5 Stationary and Nonstationary Random Processes	59
3.6 Statistics (Expected Values) of a Random Process	62
3.7 Some Useful Classes of Random Processes	73
3.8 Transformation of a Random Process	83
3.9 Summary	90
Bibliography	90

Information is knowledge or data about the states of a random variable. Information theory allows prediction and estimation through modelling the history of the dependencies of the state sequence of a random process. For example, the history of fluctuations of random variables, such as weather, the demand on a cellular mobile system or stock market prices may be used to obtain a finite-state model of these random variables. In communication signal processing, an information model is essentially a probability model of the transitions between the states of a random process and the distribution of the variable within each state of the process.

Probability models form the foundation of information theory. Information is quantified in units of 'bits' in terms of a logarithmic function of probability. Probability models are used in communications and signal processing systems to characterise and predict random signals in a diverse areas of applications such as: speech/image recognition, audio/video coding, bioengineering, weather forecasting, financial data modelling, noise reduction,

communication networks and prediction of the call demand on a service facility such as a mobile phone network.

This chapter introduces the concept of random process and probability models and explores the relationship between probability and information. The concept of *entropy* is introduced as a measure for quantification of information, and its application in Huffman coding is presented. Finally, several different forms of probability models and their applications in communication signal processing are considered.

3.1 INTRODUCTION

Probability models form the foundation of information theory. As shown in Figure 3.1, many applications of information theory, such as data and signal compression, pattern recognition, decision-making, search engines and artificial intelligence, are based on the use of probability models. As shown later in this chapter, information is measured and quantified in units of ‘bits’ in terms of a logarithmic function of probability. It would be impossible to develop advanced communication systems without the use of probability and information theory.

Information theory deals with signals that are random such as text, speech, image, noise and time series. Indeed, a signal cannot convey information without being random in the sense that a predictable signal has no information and conversely the future values of an information-bearing signal are not predictable from the past values.

The modelling, quantification and ranking of information in communication systems require appropriate mathematical tools to model the randomness of information-bearing signals, and the main tools for modelling randomness in a signal are those offered by statistics and probability theory.

In communication signal processing, an information model is essentially a probability model that models the transitions between the states of a random variable and the distribution of the variable and noise within each state of the process.

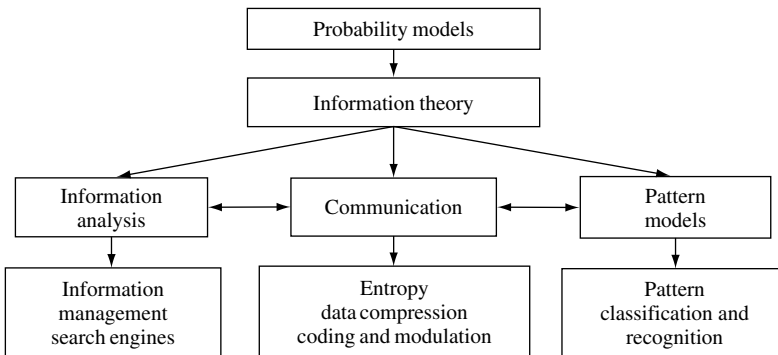


Figure 3.1 A tree-structure illustration of probability models leading to information theory and applications in information management, entropy coding and pattern recognition.

This chapter begins with a study of random processes and probability models. Probability models are used in communications and signal processing systems to characterise and predict random signals in diverse areas of applications such as speech/image recognition, audio/video coding, bioengineering, weather forecasting, financial data modelling, noise reduction, communication networks and prediction of the call demand on a service facility such as a mobile phone network.

The concepts of randomness, information and entropy are introduced and their close relationships explored. A random process can be completely described in terms of a probability model, but may also be characterised with relatively simple statistics, such as the mean, the correlation and the power spectrum. We study stationary, nonstationary and finite-state processes. We consider some widely used classes of random processes, and study the effect of filtering or transformation of a signal on its probability distribution. Finally, several applications of probability models in communication signal processing are considered.

3.2 RANDOM SIGNALS

Signals, in terms of one of their most fundamental characteristics, can be classified into two broad categories:

- (1) *deterministic* signals such as sine waves that on their own have no information content but can be modulated by a random information-bearing signal;
- (2) *Random* signals, such as speech and image, that have information.

In each class, a signal may be continuous or discrete in time, may have continuous-valued or discrete-valued amplitudes and may be one-dimensional or multidimensional.

A deterministic signal has a predetermined trajectory in time and/or space. The exact fluctuations of a deterministic signal can be described in terms of a function of time, and its exact value at any time is predictable from the functional description and the past history of the signal. For example, a sine wave $x(t)$ can be modelled, and accurately predicted either by a second-order linear predictive model or by the more familiar equation $x(t) = A \sin(2\pi ft + \phi)$. Note that a deterministic signal carries no information other than its own shape and parameters. Deterministic signals are a theoretical concept.

Random signals have unpredictable fluctuations; hence it is not possible to formulate an equation that can predict the *exact* future value of a random signal. Most signals of interest such as speech, music and noise are at least partly random. The concept of randomness is closely associated with the concepts of information, bandwidth and noise. *For a signal to have a capacity to convey information, it must have a degree of randomness: a predictable signal conveys no information.* Therefore the random part of a signal is either the information content of the signal, or noise, or a mixture of information and noise. In telecommunications it is a waste of resources such as time, power and bandwidth to retransmit the predictable part of a signal. Hence signals are randomised (de-correlated) before transmission.

Although a random signal is not predictable, it often exhibits a set of well-defined statistical values such as maximum, minimum, mean, median, variance, power spectrum and higher-order statistics. A random process is described in terms of its statistics, and most completely in terms of a probability model from which its statistics can be calculated.

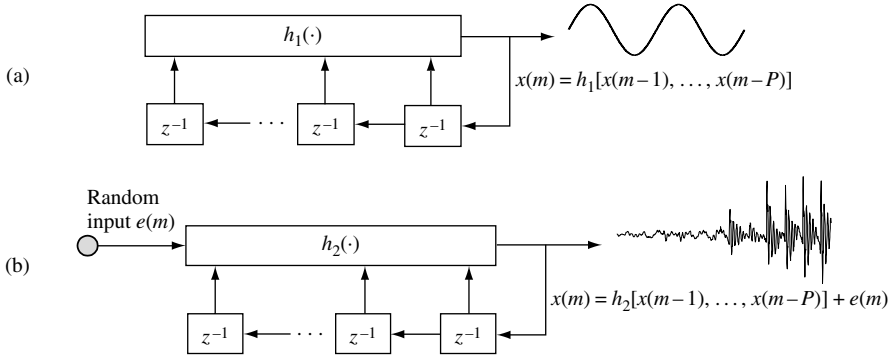


Figure 3.2 Illustration of deterministic and random signal models: (a) a deterministic signal model; (b) a random signal model.

Example 3.1: A Deterministic Signal Model

Figure 3.2(a) shows a model of a deterministic discrete-time signal. The model generates an output signal, $x(m)$, from P past samples as

$$x(m) = h_1[x(m-1), x(m-2), \dots, x(m-P)] + \delta(m) \quad (3.1)$$

where the function h_1 may be a linear or a nonlinear model and $\delta(m)$ is a delta function that acts as an initial ‘kick-start’ impulse input. Note that there is no sustained input. A functional description of the model h_1 , together with the P initial sample values, is all that is required to generate or predict the future values of the signal $x(m)$. For example for a digital sinusoidal generator (i.e. a digital oscillator), Equation (3.1) becomes

$$x(m) = a_1 x(m-1) + a_2 x(m-2) + \delta(m) \quad (3.2)$$

where the parameter $a_1 = 2 \cos(2\pi F_0/F_s)$ determines the oscillation frequency F_0 of the sinusoid at a sampling frequency of F_s .

Example 3.2: A Random Signal Model

Figure 3.2(b) is a model for a random signal given by

$$x(m) = h_2[x(m-1), x(m-2), \dots, x(m-P)] + e(m) \quad (3.3)$$

where the random input $e(m)$ models the part of the signal $x(m)$ that is unpredictable, and the function h_2 models the part of the signal that is correlated with and predictable from the past samples. For example, a narrowband, second-order autoregressive process can be modelled as

$$x(m) = a_1 x(m-1) + a_2 x(m-2) + e(m) \quad (3.4)$$

where the choice of the model parameters a_1 and a_2 will determine the centre frequency and the bandwidth of the process.

3.2.1 RANDOM AND STOCHASTIC PROCESSES

A random process generates random signals. The term ‘stochastic process’ is broadly used to describe a random process that generates *sequential* random signals such as sequence of speech samples, a video sequence, a sequence of noise samples, a sequence of stock market fluctuations or a DNA sequence. In signal processing terminology, a random or stochastic process is also a probability model of a class of random signals, e.g. Gaussian process, Markov process, Poisson process, binomial process or multinomial process.

In this chapter, we are mainly concerned with discrete-time random processes that may occur naturally or may be obtained by sampling a continuous-time band-limited random process. The term ‘discrete-time stochastic process’ refers to a class of discrete-time random signals, $X(m)$, characterised by a probabilistic model. Each realisation of a discrete-time stochastic process, $X(m)$, may be indexed in time and space as $x(m, s)$, where m is the discrete time index, and s is an integer variable that designates a space index to each realisation of the process.

3.2.2 THE SPACE OF A RANDOM PROCESS

The collection of all realisations of a random process is known as the space, or the ensemble, of the process. For an illustration, consider a random noise process over a communications network, as shown in Figure 3.3. The noise on each telephone line fluctuates randomly with time, and may be denoted $n(m, s)$, where m is the discrete-time index and s denotes the line index. The collection of noise on different lines forms the space of the noise process, denoted $N(m) = \{n(m, s)\}$, where $n(m, s)$ denotes a realisation of the noise process $N(m)$ on the line s .

The ‘true’ statistics of a random process are obtained from the averages taken over the space of many different realisations of the process. However, in many practical cases, only one or a finite number of realisations of a process are available. In Sections 3.6.8–3.6.10, we consider the ergodic processes in which time-averaged statistics, from a single realisation of a process, may be used instead of the ensemble-averaged statistics.

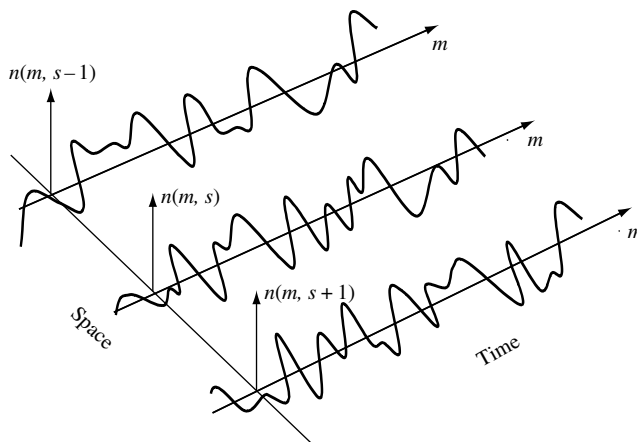


Figure 3.3 Illustration of three different realisations in the space of a random noise process, $N(m)$.

3.2.2.1 Notation

In this chapter $X(m)$, in capitals, denotes a random process, the signal $x(m, s)$ is a particular realisation of the process $X(m)$, the signal $x(m)$ is any realisation of $X(m)$, and the collection of all realisations of $X(m)$, denoted $\{x(m, s)\}$, forms the ensemble or the space of the process $X(m)$.

3.3 PROBABILITY MODELS

Probability models, initially devised to calculate the odds for the random outcomes in a chance game, provide a complete mathematical description of the distribution of a random process. In its simplest form, a probability model provides a numerical value, between 0 and 1, for the likelihood of a discrete-valued random variable assuming a particular state or value. The probability of an outcome of a variable should reflect the fraction of times that the outcome is observed to occur. It is common to quantify belief in the probability of the outcome of a process in terms of a number between 0 and 1, or in terms of its equivalent percentage.

The choice of 0 for the probability of occurrence of an infinitely improbable state or event is necessary if the laws of probability are to hold. The choice of 1, for the probability of a state or event that happens with certainty, is arbitrary, but it is a convenient and established choice.

Probability models enable the estimation of the likely values of a process from noisy or incomplete observations. As illustrated in Figure 3.4, probability models can describe random processes that are discrete-valued, continuous-valued or finite-state continuous-valued. Figure 3.4 lists the most commonly used probability models. Probability models are often functions of the statistical parameters of the random process, such as exponential functions of the mean value and covariance of the process.

At this point it is useful to define the difference between a random variable and a random process. A random variable is a variable that assumes random values such as the outcomes of a chance game or the values of a speech sample or an image pixel or the outcome of a sport match. A random process, such as a Markov process, generates random variables, usually as functions of time and space. Also a time or space series, such as a sequence of speech or an image, is often called a random process.

Consider a random process that generates a time-sequence of numbers, $x(m)$. Let $\{x(m, s)\}$ denote a collection of different sequences generated by the same process, where m denotes

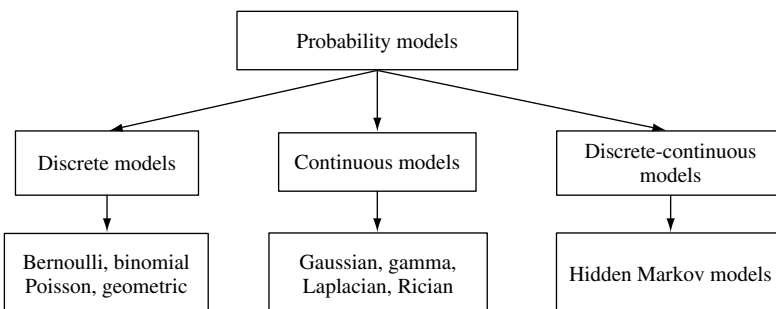


Figure 3.4 A categorisation of different classes and forms of probability models.

time and s is the sequence index. For a given time instant, m , the sample realisations of a random process $\{x(m, s)\}$ is a random variable that takes on various values across the space, s , of the process. The main difference between a random variable and a random process is that the latter generates random time/space series. Therefore the probability models used for random variables are also applied to random processes. We continue this section with the definitions of the probability functions for a random variable.

3.3.1 PROBABILITY AND RANDOM VARIABLES

Classical examples of random variables are the random outcomes in a chance process, or gambling game, such as the outcomes of throwing a coin or a pair of dice, or dealing cards in a game.

The space of a random variable is the collection of all the values, or outcomes, that the variable can assume. The space of a random variable can be partitioned, according to some criteria, into a number of subspaces. A subspace is a collection of signal values with a common attribute, such as a cluster of closely spaced samples, or the collection of samples with their values within a given band of values, for example the percentage of students attaining a certain grade in examinations.

$$P(A) = \frac{N_A}{\sum_{\text{All events } i} N_i} \quad (3.5)$$

Each subspace is called an event, and the probability of an event A , $P(A)$, is the ratio of the number of observed outcomes from the space of A , N_A , divided by the total number of observations:

From Equation (3.5), it is evident that the sum of the probabilities of all likely events in an experiment is 1:

$$\sum_{\text{All events } A} P(A) = 1 \quad (3.6)$$

Example 3.3

The space of two discrete numbers obtained as outcomes of throwing a pair of dice is shown in Figure 3.5. This space can be partitioned in different ways; for example, the two subspaces A and B shown in Figure 3.5 are associated with the pair of numbers that in the case of subspace A add up to a value of greater than 8, and in the case of subspace B add up to value of less than or equal to 8. In this example, assuming that the dice are not loaded, all numbers are equally likely and the probability of each event is proportional to the total number of outcomes in the space of the event, as shown in the figure.

3.3.2 PROBABILITY MASS FUNCTION

For a discrete random variable X that can assume values from a finite set of N numbers or symbols $\{x_1, x_2, \dots, x_N\}$, each outcome, x_i , may be considered an event and assigned a

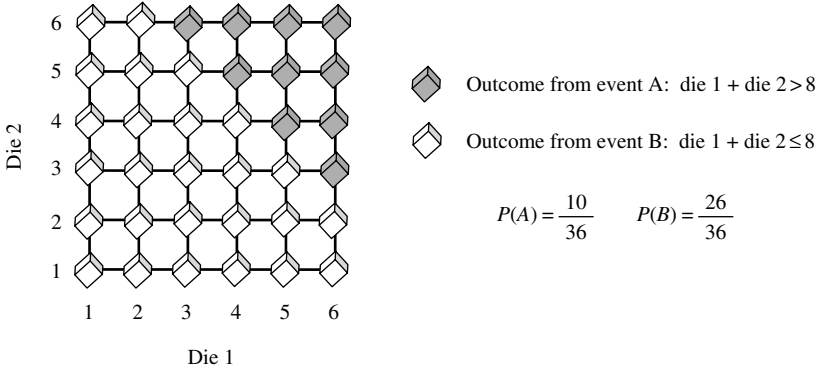


Figure 3.5 A two-dimensional representation of the outcomes of two dice, and the subspaces associated with the events corresponding to the sum of the dice being greater than 8, or less than or equal to 8. $P(A) + P(B) = 1$.

probability of occurrence. For example, if the variable is the outcome of tossing a coin, then the outcomes are heads (H) and tails (T), hence $X = \{H, T\}$ and $P(X = H) = P(X = T) = 0.5$.

The probability that a discrete-valued random variable X takes on a value of x_i , $P(X = x_i)$, is called the *probability mass function (pmf)*. For two such random variables X and Y , the probability of an outcome in which X takes on a value of x_i and Y takes on a value of y_j , $P(X = x_i, Y = y_j)$, is called the *joint probability mass function*.

The joint pmf can be described in terms of the conditional and the marginal probability mass functions as

$$\begin{aligned}
 P_{X,Y}(x_i, y_j) &= P_{Y|X}(y_j|x_i)P_X(x_i) \\
 &= P_{X|Y}(x_i|y_j)P_Y(y_j)
 \end{aligned}
 \tag{3.7}$$

where $P_{Y|X}(y_j|x_i)$ is the *conditional* probability of the random variable Y taking on a value of y_j conditioned on the variable X having taken a value of x_i , and the so-called *marginal* pmf of X is obtained as

$$\begin{aligned}
 P_X(x_i) &= \sum_{j=1}^M P_{X,Y}(x_i, y_j) \\
 &= \sum_{j=1}^M P_{X|Y}(x_i|y_j)P_Y(y_j)
 \end{aligned}
 \tag{3.8}$$

where M is the number of values, or outcomes, in the space of the discrete random variable Y .

3.3.2.1 Bayes’s Rule

Assume we wish to find the probability that a random variable, X , takes a value of x_i given that a related variable, Y , has taken a value of y_j . From Equations (3.7) and (3.8), we have

Bayes's rule, for the conditional probability mass function, given by

$$\begin{aligned}
 P_{X|Y}(x_i|y_j) &= \frac{1}{P_Y(y_j)} P_{Y|X}(y_j|x_i) P_X(x_i) \\
 &= \frac{P_{Y|X}(y_j|x_i) P_X(x_i)}{\sum_{i=1}^M P_{Y|X}(y_j|x_i) P_X(x_i)}
 \end{aligned}
 \tag{3.9}$$

Bayes's rule forms the foundation of probabilistic estimation and classification theory, introduced in Chapter 4.

Example 3.4: Probability of the Sum of Two Random Variables

Figure 3.6(a) shows the pmf of a die. Now, let the variables (x, y) represent the outcomes of throwing a pair of dice. The probability that the sum of the outcomes of throwing two dice is equal to A , is given by

$$P(x + y = A) = \sum_{i=1}^6 P(x = i) P(y = A - i)
 \tag{3.10}$$

The pmf of the sum of two dice is plotted in Figure 3.6(b). Note from Equation (3.10) that the probability of the sum of two random variables is the convolution sum of the probability functions of the individual variables.

3.3.3 PROBABILITY DENSITY FUNCTION

A continuous-valued random variable can assume an infinite number of values, in even a very small range of values, and hence the probability that it takes on any given value is infinitely small and vanishes to zero.

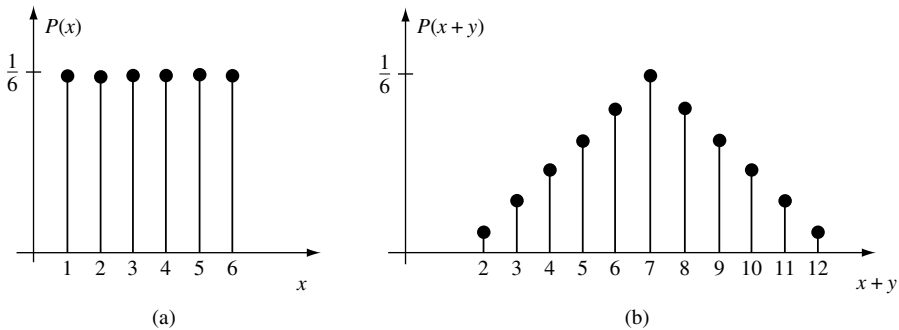


Figure 3.6 The probability mass function (pmf) of (a) a die, and (b) the sum of a pair of dice.

For a continuous-valued random variable X the cumulative distribution function (cdf) is defined as the probability that the outcome is less than x :

$$F_X(x) = \text{Prob}(X \leq x) \quad (3.11)$$

where $\text{Prob}(\cdot)$ denotes probability. The probability that a random variable X takes on a value within a range of values Δ centred on x can be expressed as

$$\begin{aligned} \frac{1}{\Delta} \text{Prob}(x - \Delta/2 \leq X \leq x + \Delta/2) &= \frac{1}{\Delta} [\text{Prob}(X \leq x + \Delta/2) - \text{Prob}(X \leq x - \Delta/2)] \\ &= \frac{1}{\Delta} [F_X(x + \Delta/2) - F_X(x - \Delta/2)] \end{aligned} \quad (3.12)$$

Note that both sides of Equation (3.12) are divided by Δ . As the interval Δ tends to zero, we obtain the probability density function (pdf) as

$$f_X(x) = \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} [F_X(x + \Delta/2) - F_X(x - \Delta/2)] = \frac{\partial F_X(x)}{\partial x} \quad (3.13)$$

Since $F_X(x)$ increases with x , the pdf of x is a non-negative-valued function, i.e. $f_X(x) \geq 0$. The integral of the pdf of a random variable X in the range $\pm\infty$ is unity:

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad (3.14)$$

The conditional and marginal probability functions and the Bayes' rule of Equations (3.7)–(3.9) also apply to pdfs of continuous-valued variables.

3.3.4 PROBABILITY DENSITY FUNCTIONS OF RANDOM PROCESSES

The probability models obtained for random variables can be applied to random processes such as time series, speech and images. For a continuous-valued random process, $X(m)$, the simplest probabilistic model is the univariate pdf $f_{X(m)}(x)$, which is the pdf of a sample from the random process $X(m)$ taking on a value of x . A bi-variate pdf, $f_{X(m)X(m+n)}(x_1, x_2)$, describes the pdf of two samples of the process X at time instants m and $m+n$ taking on the values x_1 and x_2 , respectively.

In general, an M -variate pdf, $f_{X(m_1)X(m_2)\dots X(m_M)}(x_1, x_2, \dots, x_M)$, describes the pdf of a vector of M samples of a random process taking on specific values at specific time instants. For an M -variate pdf, we can write

$$\int_{-\infty}^{\infty} f_{X(m_1)\dots X(m_M)}(x_1, \dots, x_M) dx_M = f_{X(m_1)\dots X(m_{M-1})}(x_1, \dots, x_{M-1}) \quad (3.15)$$

and the sum of the pdfs of all realisations of a random process is unity, i.e.

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{X(m_1) \cdots X(m_M)}(x_1, \dots, x_M) dx_1 \dots dx_M = 1 \quad (3.16)$$

The probability of the value of a random process at a specified time instant may be conditioned on the value of the process at some other time instant, and expressed as a conditional probability density function as

$$f_{X(m)|X(n)}(x_m | x_n) = \frac{f_{X(n)|X(m)}(x_n | x_m) f_{X(m)}(x_m)}{f_{X(n)}(x_n)} \quad (3.17)$$

Equation (3.17) is the Bayes' rule. If the outcome of a random process at any time is independent of its outcomes at other time instants, then the random process is uncorrelated. For an uncorrelated process a multivariate pdf can be written in terms of the products of univariate pdfs as

$$f_{[X(m_1) \cdots X(m_M)]|X(n_1) \cdots X(n_N)}(x_{m_1}, \dots, x_{m_M} | x_{n_1}, \dots, x_{n_N}) = \prod_{i=1}^M f_{X(m_i)}(x_{m_i}) \quad (3.18)$$

Discrete-valued random processes can only assume values from a finite set of allowable numbers, $[x_1, x_2, \dots, x_n]$. An example is the output of a binary digital communication system that generates a sequence of 1s and 0s. Discrete-time, discrete-valued, stochastic processes are characterised by multivariate probability mass functions (pmf) denoted

$$P_{[x(m_1) \cdots x(m_M)]}[x(m_1) = x_i, \dots, x(m_M) = x_k] \quad (3.19)$$

The probability that a discrete random process $X(m)$ takes on a value of x_m at a time instant m can be conditioned on the process taking on a value x_n at some other time instant n , and expressed in the form of a conditional pmf as

$$P_{X(m)|X(n)}(x_m | x_n) = \frac{P_{X(n)|X(m)}(x_n | x_m) P_{X(m)}(x_m)}{P_{X(n)}(x_n)} \quad (3.20)$$

and for a statistically independent process we have

$$P_{[X(m_1) \cdots X(m_M)]|X(n_1) \cdots X(n_N)}(x_{m_1}, \dots, x_{m_M} | x_{n_1}, \dots, x_{n_N}) = \prod_{i=1}^M P_{X(m_i)}[X(m_i) = x_{m_i}] \quad (3.21)$$

3.3.4.1 Histogram

A histogram is a graph that shows the number of times the value of a random variable occurs in each uniform interval in a given set of observations of the variables. Given a set of observations of a random variable, the range of values of the variable is divided into N equal width bins and the number of times that the variable falls within each bin is calculated. A histogram shows the form of the probability distribution of a variable as indicated by a set of observations. Figure 3.7 shows the histogram and the probability model of Gaussian signal.

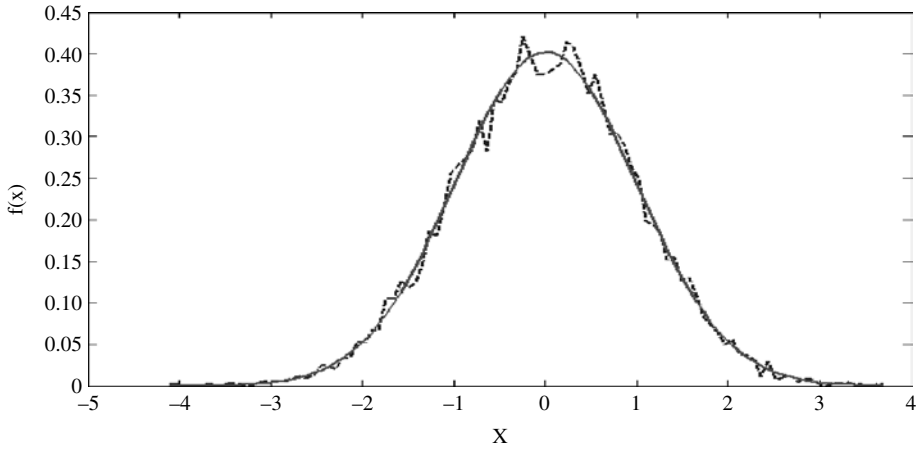


Figure 3.7 Histogram (dashed line) and probability model of a Gaussian signal.

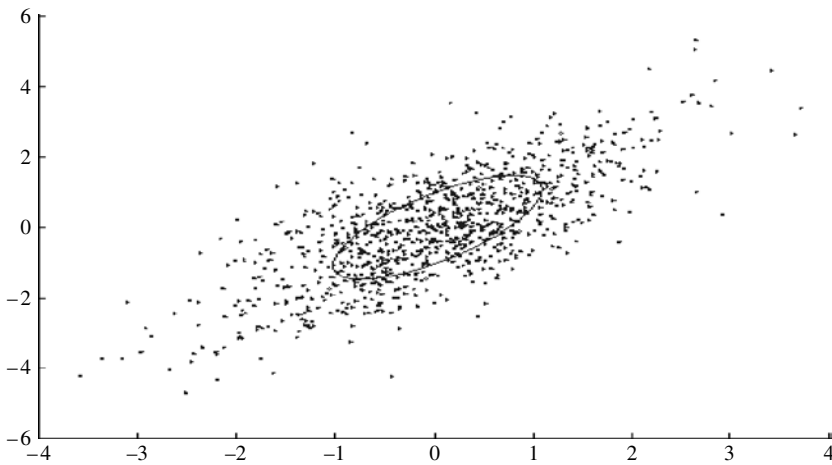


Figure 3.8 The scatter plot of a two-dimensional Gaussian distribution.

Figure 3.8 shows the scatter plot of a two-dimensional Gaussian process superimposed on an ellipse which represents the standard deviation contour.

3.4 INFORMATION MODELS

Information is knowledge regarding the states of a random variable. Information is discrete in nature and can be represented in a binary format in terms M of states of a variable. The states of an information-bearing variable may be arranged in a binary tree structure, as shown later in Example 3.10.

We shall show in this section that information is measured in terms of units of bits. One bit of information is equivalent to two equal-probability states. Note that the observation from which information is obtained may be continuous-valued or discrete-valued.

The information conveyed by a random process is associated with its state sequence. Examples are the states of weather, health, emotion, market share price indices, communication symbols and DNA or protein sequences.

The concepts of information, randomness and probability models are closely related. For a signal to have information it must satisfy two conditions:

- (1) it must possess two or more states or values;
- (2) it must move between the states in a random manner.

For example, the outcome of tossing a coin is an unpredictable binary state (heads/tails) event, a digital communications system with N -bit codewords has 2^N states and the outcome of a weather forecast can be one or more of the following states: {sun, cloud, cold, warm, hot, rain, snow, storm, etc.}.

Random processes are modelled with probability functions. It is therefore natural that information is modelled as a function of probability. The expected (average) *information content* of a state x_i of a random variable is quantified as

$$I(x_i) = -P_X(x_i) \log P_X(x_i) \text{ bits} \quad (3.22)$$

where the base of logarithm is 2. For a binary source the information conveyed by the two states $[x_1, x_2]$ can be described as

$$\begin{aligned} H(X) &= I(x_1) + I(x_2) \\ &= -P(x_1) \log P(x_1) - P(x_2) \log P(x_2) \end{aligned} \quad (3.23)$$

Alternatively $H(X)$ in Equation (3.23) can be written as

$$H(X) = -P(x_i) \log P(x_i) - [1 - P(x_i)] \log [1 - P(x_i)] \quad i = 1 \text{ or } 2 \quad (3.24)$$

Note from Figure 3.9 that the information content of a variable has a value of 0 for an event whose probability is 0, i.e. an impossible event, and its value is also 0 for an event that happens with probability of 1.

3.4.1 ENTROPY

Entropy gives a measure of the quantity of the information content of a random variable in terms of the minimum number of bits per symbol required to encode the variable. Entropy can be used to calculate the theoretical minimum capacity or bandwidth required for the storage or transmission of an information source such as text, image, music, etc. In his pioneering work: 'A mathematical theory of communication' (*Bell Syst. Tech. J.*, **27**: 379–423, 623–656),

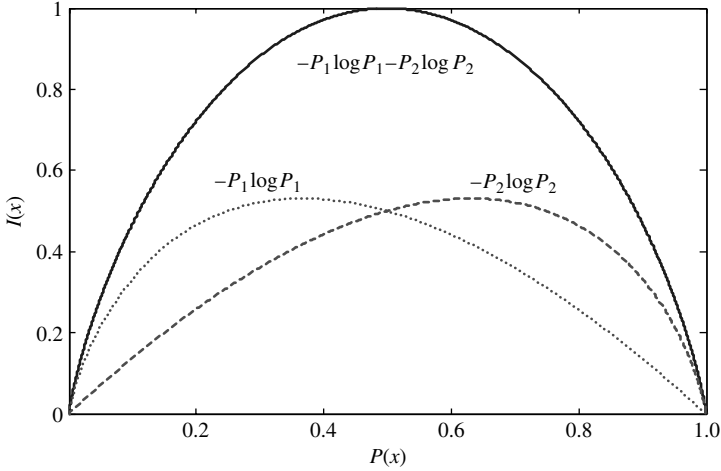


Figure 3.9 Illustration of $I(x_i)$ vs $P(x_i)$; for a binary source the maximum information content is one bit, when the two states have equal probability of 0.5. Note: $P(x_1) = P_1$ and $P(x_2) = P_2$.

C.E. Shannon derived the entropy measure, H , as a function that satisfies the following conditions:

- (1) entropy H should be a continuous function of P_i ;
- (2) for $P_i = 1/M$, H should be a monotonically increasing function of M ;
- (3) if the communication symbols are broken into two (or more) sets the entropy of the original set should be equal to the probability-weighted sum of the entropy of the subsets.

$$H(X) = - \sum_{i=1}^M P(x_i) \log P(x_i) \text{ bits} \quad (3.25)$$

Consider a random variable X with M states $[x_1, x_2, \dots, x_M]$ and state probabilities $[p_1, p_2, \dots, p_M]$ where $P_X(x_i) = p_i$; the entropy of X is defined as where the base of the logarithm is 2. The log function has several useful properties. Log 1 is 0, which is a useful mapping, as an event with probability of 1 has 0 information. Furthermore, with the use of logarithm, the addition of a binary state to M existing binary states doubles the number of choices from 2^M to 2^{M+1} but increases the logarithm of the number of states by one. Hence 1 bit of information corresponds to two equal probability states, 2 bits correspond to four states, and so on. The base 2 of the logarithm reflects the binary nature of information. Information is discrete and can be represented as a set of binary symbols.

Entropy is measured in units of bits. It is bounded as

$$0 \leq H(X) \leq \log_2 M \quad (3.26)$$

where $H(X) = 0$ if one symbol, x_i , has a probability of 1 and all other symbols have probabilities of 0, and M denotes the number of symbols in the set X . *The entropy of a set*

attains a maximum value of $\log_2 M$ bits for a uniformly distributed M -valued variable with each outcome having a probability of $1/M$.

Entropy gives the minimum number of bits per symbol required for binary coding of different values of a random variable X . This theoretical minimum is usually approached by encoding N samples of the process simultaneously with K bits where the number of bits per sample $K/N \geq H(X)$. As N becomes large, for an efficient coder the number of bits per sample K/N approaches the entropy $H(X)$ of X (see Huffman code in Section 3.4.3).

3.4.1.1 Shannon's Source Coding Theorem

N independent identically distributed (IID) random variables each with entropy H can be compressed into more than NH bits with negligible loss of quality as $N \rightarrow \infty$.

Example 3.5: Entropy of the English Alphabet

Calculate the entropy of the set of the English alphabet [A, B, C, D, . . . , Z], assuming that all letters are equally likely. Hence, calculate the theoretical minimum number of bits required to code a text file of 2000 words with an average of five letters per word.

For the English alphabet the number of symbols $N = 26$, and, assuming that all symbols are equally likely, the probability of each symbol becomes $p_i = 1/26$. Using Equation (3.25) we have

$$H(X) = - \sum_{i=1}^{26} \frac{1}{26} \log_2 \frac{1}{26} = 4.7 \text{ bits} \quad (3.27)$$

The total number of bits for encoding 2000 words = $4.7 \times 2000 \times 5 = 47$ kbits. Note that different letter type cases (upper case, lower case, etc.) and symbols (!, ?, etc) are not taken into account and also note that the actual distribution of the letters is non-uniform, resulting in an entropy of less than 4.7 bits/symbol.

Example 3.6: Entropy of the English Alphabet Using Estimates of Probabilities of Letters of the Alphabet

Use the set of probabilities of alphabet shown in Figure 3.10: $P(A) = 0.0856$, $P(B) = 0.0139$, $P(C) = 0.0279$, $P(D) = 0.0378$, $P(E) = 0.1304$, $P(F) = 0.0289$, $P(G) = 0.0199$, $P(H) = 0.0528$, $P(I) = 0.0627$, $P(J) = 0.0013$, $P(K) = 0.0042$, $P(L) = 0.0339$, $P(M) = 0.0249$, $P(N) = 0.0707$, $P(O) = 0.0797$, $P(P) = 0.0199$, $P(Q) = 0.0012$, $P(R) = 0.0677$, $P(S) = 0.0607$, $P(T) = 0.1045$, $P(U) = 0.0249$, $P(V) = 0.0092$, $P(W) = 0.0149$, $P(X) = 0.0017$, $P(Y) = 0.0199$, $P(Z) = 0.0008$.

The entropy of this set is given by

$$H(X) = - \sum_{i=1}^{26} P_i \log_2 P_i = 4.13 \text{ bits/symbol} \quad (3.28)$$

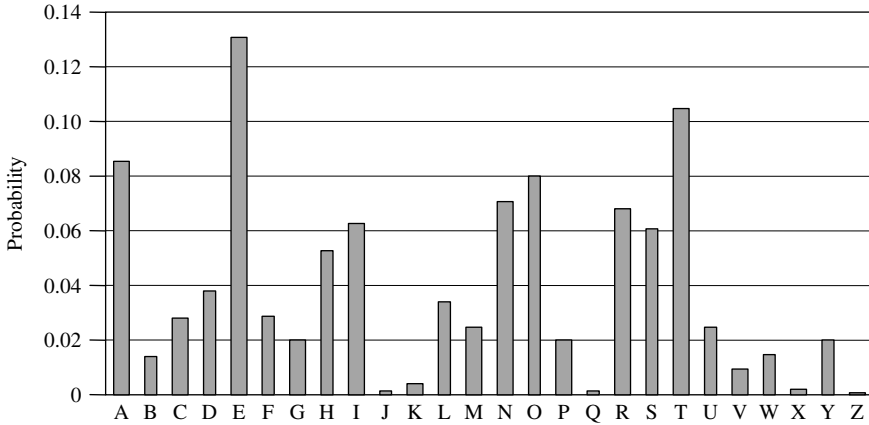


Figure 3.10 The probability of the English alphabet, A–Z.

Example 3.7: Entropy of the English Phonemes

Spoken English is constructed from about 40 basic acoustic symbols, known as phonemes (or phonetic units), and these are used to construct words, sentences, etc. For example the word ‘signal’ is transcribed in phonemic form as ‘s iy g n aa l’. Assuming that all phonemes are equiprobable, the average speaking rate is 120 words per minute and the average word has four phonemes, calculate the minimum number of bits per second required to encode speech at the average speaking rate.

For speech, $N = 40$, assume $P_i = 1/40$. The entropy of phonemes is given by

$$H(X) = - \sum_{i=1}^{40} \frac{1}{40} \log_2 \frac{1}{40} = 5.3 \text{ bits/symbol} \quad (3.29)$$

$$\begin{aligned} \text{Number of bits/s} &= (120/60 \text{ words per second}) \times (4 \text{ phonemes per word}) \\ &\times (5.3 \text{ bits per phoneme}) = 43.4 \text{ bps} \end{aligned}$$

Note that the actual distribution of phonemes is non-uniform, resulting in an entropy of less than 5.3 bits. Furthermore, the above calculation ignores the information (and the hence entropy) in speech, due to contextual variations of phonemes, speaker identity, accent, pitch intonation and emotion signals.

3.4.2 MUTUAL INFORMATION

Consider two correlated random variables X and Y ; the conditional entropy of X given Y is defined as

$$H(X|Y) = - \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} P(x_i, y_j) \log P(x_i|y_j) \quad (3.30)$$

$H(X|Y)$ is equal to $H(X)$ if Y is uncorrelated with X and is equal to 0 if Y has the same information as X . The information that the variable Y contains about the variable X is given by

$$I(X; Y) = H(X) - H(X|Y) \tag{3.31}$$

Substituting Equation (3.30) in Equation (3.31) and also using the relation

$$H(X) = - \sum_{i=1}^{M_x} P(x_i) \log P(x_i) = - \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} P(x_i, y_j) \log P(x_i) \tag{3.32}$$

yields

$$I(X; Y) = \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \tag{3.33}$$

Note from Equation (3.33) that $I(X; Y) = I(Y; X)$, that is the information that Y has about X is the same as that X has about Y , hence $I(X; Y)$ is *mutual information*. As shown next, mutual information has a minimum of 0, $I(X; Y) = 0$, for independent variables and a maximum of $I(X; Y) = H(X) = H(Y)$ when X and Y have identical information.

Example 3.8: Upper and Lower Bounds on Mutual Information

Obtain the bounds on mutual information of two random variables X and Y . The upper bound is given when X and Y contain identical information; in this case, substituting $P(x_i, y_j) = P(x_i)$ and $P(y_j) = P(x_i)$ in Equation (3.33) and assuming that each x_i has a mutual relation with only one y_j , we have

$$I(X; Y) = \sum_{i=1}^{M_x} P(x_i) \log \frac{P(x_i)}{P(x_i)P(x_i)} = H(X) \tag{3.34}$$

The lower bound is given by the case when X and Y are independent:

$$I(X; Y) = \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} P(x_i)P(y_j) \log \frac{P(x_i)P(y_j)}{P(x_i)P(y_j)} = 0 \tag{3.35}$$

Example 3.9

Show that the *mutual entropy* of two independent variables X and Y is additive. Assume X and Y are M -valued and N -valued variables, respectively. The entropy of two random variables is given by

$$H(X, Y) = \sum_{i=1}^M \sum_{j=1}^N P_X(x_i, y_j) \log \frac{1}{P_{XY}(x_i, y_j)} \tag{3.36}$$

Substituting $P(x_i, y_j) = P(y_j)P(x_i)$ in Equation (3.34) yields

$$\begin{aligned}
 H(X, Y) &= \sum_{i=1}^M \sum_{j=1}^N P_X(x_i, y_j) \log \frac{1}{P_{XY}(x_i, y_j)} \\
 &= - \sum_{i=1}^M \sum_{j=1}^N P_X(x_i) P_Y(y_j) \log P_X(x_i) - \sum_{i=1}^M \sum_{j=1}^N P_X(x_i) P_Y(y_j) \log P_Y(y_j) \\
 &= - \sum_{i=1}^M P_X(x_i) \log P_X(x_i) - \sum_{j=1}^N P_Y(y_j) \log P_Y(y_j) \\
 &= H(X) + H(Y)
 \end{aligned} \tag{3.37}$$

where we have used the following relations

$$\sum_{j=1}^N P_Y(y_j) P_X(x_i) \log P_X(x_i) = P_X(x_i) \log P_X(x_i) \sum_{j=1}^N P_Y(y_j) = P_X(x_i) \log P_X(x_i) \tag{3.38}$$

and for two independent variables

$$\log[1/P_{XY}(x_i, y_j)] = - \log P_X(x_i) - \log P_Y(y_j) \tag{3.39}$$

3.4.3 ENTROPY CODING

Entropy gives the minimum number of bits required to encode an information source. This theoretical minimum may be approached by encoding N samples of a signal simultaneously with K bits where $K/N \geq H(X)$. As N becomes large, for an efficient coder K/N approaches the entropy $H(X)$ of X . The efficiency of a coding scheme in terms of its entropy is defined as $H(X)/N$. When $N = H(X)$ then the entropy coding efficiency of the code is $H(X)/N = 1$ or 100%.

The average number of bits per symbol, also known as the ‘average code length’, CL , can be expressed as

$$CL = \sum_{i=1}^M P(x_i) L(x_i) \tag{3.40}$$

where $L(x_i)$ is the length of the binary codeword used to encode symbol x_i and $P(x_i)$ is the probability of x_i . A comparison of Equation (3.40) with the entropy Equation (3.25) shows that, for an optimal code, $L(x_i)$ is $-\log_2(x_i)$. The aim of the design of minimum length code is that the *average* code length should approach the entropy.

The simplest method to encode an M -valued variable is to use a fixed-length code that assigns N binary digits to each of the M values with $N = \text{Nint}(\log_2 M)$, where Nint denotes the nearest integer round-up function. When the source symbols are *not equally probable*, a more efficient method is *entropy encoding*.

Entropy coding is a variable-length coding method which assigns codewords of variable lengths to communication alphabet symbols $[x_k]$ such that the more probable symbols, which

occur more frequently, are assigned shorter codewords and the less probable symbols, which happen less frequently, are assigned longer code words. An example of such a code is Morse Code, which dates back to the nineteenth century. Entropy coding can be applied to the coding of music, speech, image, text and other forms of communication symbols. If the entropy coding is ideal, the bit rate at the output of a uniform M -level quantiser can be reduced by an amount of $\log_2 M - H(X)$ compared with fixed-length coding.

3.4.3.1 Huffman Code

A simple and efficient form of entropy coding is the *Huffman code*, which creates a set of prefix codes (no code is part of the beginning of another code) for a given text. Huffman devised his code while he was a student at Massachusetts Institute of Technology. The ease with which Huffman codes can be created and used makes this code a popular tool for data compression. In one form of Huffman tree code, illustrated in Figure 3.11, the symbols are arranged in decreasing order of probabilities in a column. The two symbols with the lowest probabilities are combined by drawing a straight line to connect them. This combination is combined with the next symbol and the procedure is repeated to cover all symbols. Binary codewords are assigned by moving from the root of the tree at the right-hand side to the left in the tree and assigning a 1 to the lower branch and a 0 to the upper branch where all pairs of symbols have been combined.

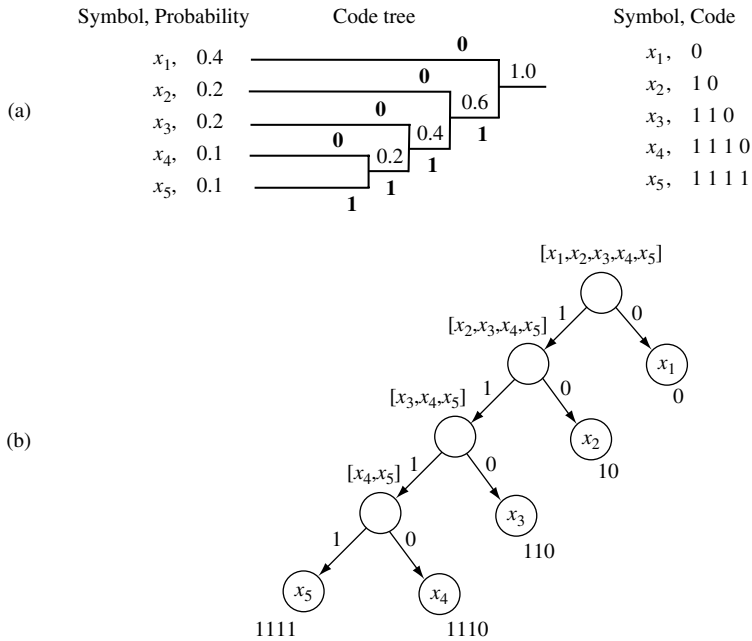


Figure 3.11 (a) Illustration of Huffman coding tree; the source entropy is 2.1219 bits/sample and Huffman code gives 2.2 bits/sample. (b) Alternative illustration of the Huffman tree.

Example 3.10

Given five symbols, x_1, x_2, \dots, x_5 , with probabilities of $P(x_1) = 0.4, P(x_2) = P(x_3) = 0.2$ and $P(x_4) = P(x_5) = 0.1$, design a binary variable length code for this source.

The entropy of X is $H(X) = 2.122$ bits/symbol. Figure 3.11 illustrates the design of a Huffman code for this source. For this tree we have:

$$\begin{aligned} \text{The average codeword length} &= 1 \times 0.4 + 2 \times 0.2 + 4 \times 0.1 + 4 \times 0.1 + 3 \times 0.2 \\ &= 2.2 \text{ bits/symbol} \end{aligned}$$

The average codeword length of 2.2 bits/symbol is close to the entropy of 2.1219 bits/symbol. We can get closer to the minimum average codeword length by encoding pairs of symbols or blocks of more than two symbols at a time (with added complexity). The Huffman code has a useful *prefix condition* property whereby no codeword is a prefix or an initial part of another codeword. Thus codewords can be readily concatenated (in a comma-free fashion) and be uniquely (unambiguously) decoded.

Figure 3.12 illustrates an alternative Huffman code tree created by a series of successive binary division of the symbols into two sets with as near set probabilities as possible. At each node the set splitting process is repeated until the leaf node containing a single symbol is reached. Binary bits of 0 and 1 are assigned to tree branches as shown. Each end-node with a single symbol represents a leaf node and is assigned a binary code which is read from the top (root) node to the leaf node.

Example 3.11

Given four symbols, x_1, x_2, x_3 and x_4 , with probabilities of $P(x_1) = 0.4, P(x_2) = 0.3, P(x_3) = 0.2$ and $P(x_4) = 0.1$, design a variable length coder to encode two symbols at a time.

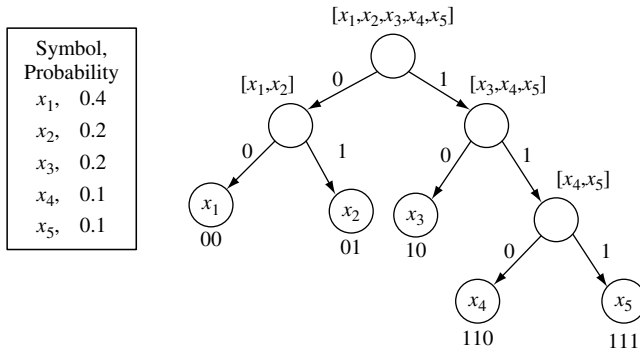


Figure 3.12 A binary Huffman coding tree. From the top node at each stage the set of symbols is divided into two sets with as near set probability (maximum entropy) as possible. Each end-node with a single symbol represents a leaf and is assigned a binary code which is read from the top node to the leaf node.

The entropy of the four symbols is obtained from Equation (3.25) as 1.8464 bits/symbol. Assuming the symbols are independent, the probability of 16 pairs of symbols can be written as in the following matrix:

$$\begin{aligned}
 P(x_1, x_1) &= 0.16, P(x_1, x_2) = 0.12, P(x_1, x_3) = 0.08, P(x_1, x_4) = 0.04 \\
 P(x_2, x_1) &= 0.12, P(x_2, x_2) = 0.09, PP(x_2, x_3) = 0.06, P(x_2, x_4) = 0.03 \\
 P(x_3, x_1) &= 0.08, P(x_3, x_2) = 0.06, PP(x_3, x_3) = 0.04, P(x_3, x_4) = 0.02 \\
 P(x_4, x_1) &= 0.04, P(x_4, x_2) = 0.03, PP(x_4, x_3) = 0.02, P(x_4, x_4) = 0.01
 \end{aligned}$$

The 16 pairs of symbols and their probabilities can be used in a Huffman tree code similar to those illustrated in Figures 3.11 and 3.12.

3.5 STATIONARY AND NONSTATIONARY RANDOM PROCESSES

Although the amplitude of a signal fluctuates with time m , the parameters of the process that generates the signal may be time-invariant (stationary) or time-varying (nonstationary). Examples of nonstationary processes are speech and music, whose loudness and spectral composition change continuously as the speaker and/or instrument generate various sounds.

A process is stationary if the parameters of the probability model of the process are time-invariant; otherwise it is nonstationary (Figure 3.13). The stationary property implies that all the statistical parameters, such as the mean, the variance, the power spectral composition and the higher-order moments of the process, are constant. In practice, there are various degrees of stationarity: it may be that one set of the statistics of a process is stationary whereas another set is time-varying. For example, a random process may have a time-invariant mean, but a time-varying power.

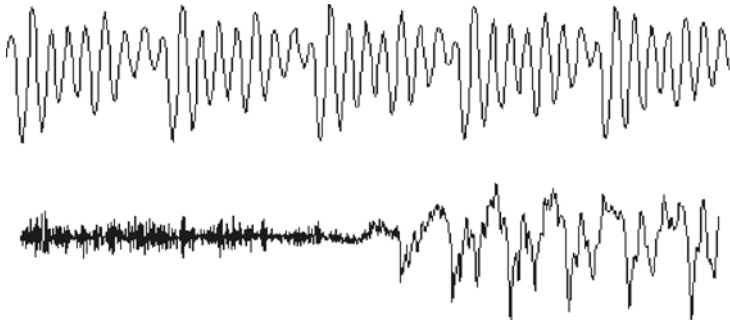


Figure 3.13 Examples of a quasistationary voiced speech (above) and a nonstationary speech composed of unvoiced and voiced segments.

Example 3.12

In this example we consider the *time-averaged* values of the mean and the power of: (a) a stationary signal, $A \sin \omega t$, and (b) a transient exponential signal, $Ae^{-\alpha t}$. The mean and power of the sinusoid, integrated over one period, are

$$\text{Mean}(A \sin \omega t) = \frac{1}{T} \int_T A \sin \omega t \, dt = 0, \quad \text{constant} \quad (3.41)$$

$$\text{Power}(A \sin \omega t) = \frac{1}{T} \int_T A^2 \sin^2 \omega t \, dt = \frac{A^2}{2}, \quad \text{constant} \quad (3.42)$$

where T is the period of the sine wave. The mean and the power of the transient signal are given by:

$$\text{Mean}(Ae^{-\alpha t}) = \frac{1}{T} \int_t^{t+T} Ae^{-\alpha \tau} \, d\tau = \frac{A}{\alpha T} (1 - e^{-\alpha T}) e^{-\alpha t}, \quad \text{time-varying} \quad (3.43)$$

$$\text{Power}(Ae^{-\alpha t}) = \frac{1}{T} \int_t^{t+T} A^2 e^{-2\alpha \tau} \, d\tau = \frac{A^2}{2\alpha T} (1 - e^{-2\alpha T}) e^{-2\alpha t}, \quad \text{time-varying} \quad (3.44)$$

In Equations (3.43) and (3.44), the signal mean and power are exponentially decaying functions of the time variable t .

Example 3.13: A Binary State Nonstationary Random Process

Consider a nonstationary signal, $y(m)$, generated by a binary-state random process described by the following equation:

$$y(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \quad (3.45)$$

where $s(m)$ is a binary-valued state-indicator variable and $\bar{s}(m)$ is the binary complement of $s(m)$. From Equation (3.45), we have

$$y(m) = \begin{cases} x_0(m) & \text{if } s(m) = 0 \\ x_1(m) & \text{if } s(m) = 1 \end{cases} \quad (3.46)$$

Let μ_{x_0} and P_{x_0} denote the mean and the power of the signal $x_0(m)$, and μ_{x_1} and P_{x_1} the mean and the power of $x_1(m)$ respectively. The expectation of $y(m)$, given the state $s(m)$, is obtained as

$$\begin{aligned} \mathcal{E}[y(m)|s(m)] &= \bar{s}(m)\mathcal{E}[x_0(m)] + s(m)\mathcal{E}[x_1(m)] \\ &= \bar{s}(m)\mu_{x_0} + s(m)\mu_{x_1} \end{aligned} \quad (3.47)$$

In Equation (3.47), the mean of $y(m)$ is expressed as a function of the state of the process at time m . The power of $y(m)$ is given by

$$\begin{aligned} \mathcal{E}[y^2(m)|s(m)] &= \bar{s}(m)\mathcal{E}[x_0^2(m)] + s(m)\mathcal{E}[x_1^2(m)] \\ &= \bar{s}(m)P_{x_0} + s(m)P_{x_1} \end{aligned} \quad (3.48)$$

Although most signals are nonstationary, the concept of a stationary process plays an important role in the development of signal processing methods. Furthermore, most nonstationary signals such as speech can be considered as approximately stationary for a short period of time. In signal processing theory, two classes of stationary processes are defined: (a) strict-sense stationary processes; and (b) wide-sense stationary processes, which is a less strict form of stationarity, in that it only requires that the first-order and second-order statistics of the process should be time-invariant.

3.5.1 STRICT-SENSE STATIONARY PROCESSES

A random process $X(m)$ is said to be strict-sense stationary if all its distributions and statistics are time-invariant. Strict-sense stationarity implies that the n th order distribution is translation-invariant for all $n = 1, 2, 3, \dots$:

$$\begin{aligned} \text{Prob}[x(m_1) \leq x_1, x(m_2) \leq x_2, \dots, x(m_n) \leq x_n] \\ = \text{Prob}[x(m_1 + \tau) \leq x_1, x(m_2 + \tau) \leq x_2, \dots, x(m_n + \tau) \leq x_n] \end{aligned} \quad (3.49)$$

where τ is any arbitrary shift along the time axis. From Equation (3.49) the statistics of a strict-sense stationary process are time-invariant. In general we have

$$\mathcal{E}[x^{k_1}(m_1), x^{k_2}(m_1 + \tau_1), \dots, x^{k_L}(m_1 + \tau_L)] = \mathcal{E}[x^{k_1}(m_2), x^{k_2}(m_2 + \tau_2), \dots, x^{k_L}(m_2 + \tau_L)] \quad (3.50)$$

where k_1, \dots, k_L are arbitrary powers. For a strict-sense stationary process, all the moments of a signal are time-invariant. The first-order moment, i.e. the mean, and the second order moments, i.e. the correlation and power spectrum, of a stationary process are given by

$$\mathcal{E}[x(m)] = \mu_x \quad (3.51)$$

$$\mathcal{E}[x(m)x(m+k)] = r_{xx}(k) \quad (3.52)$$

and

$$\mathcal{E}[|X(f, m)|^2] = \mathcal{E}[|X(f)|^2] = P_{XX}(f) \quad (3.53)$$

where μ_x , $r_{xx}(k)$ and $P_{XX}(f)$ are the mean value, the autocorrelation and the power spectrum of the signal $x(m)$, respectively, and $X(f, m)$ denotes the frequency–time spectrum of $x(m)$.

3.5.2 WIDE-SENSE STATIONARY PROCESSES

The strict-sense stationarity condition requires that all statistics of the process should be time-invariant. A less restrictive form of a stationary process is called ‘wide-sense stationarity’. A process is said to be wide-sense stationary if the mean and the autocorrelation functions (first- and second-order statistics) of the process are time-invariant:

$$\mathcal{E}[x(m)] = \mu_x \quad (3.54)$$

$$\mathcal{E}[x(m)x(m+k)] = r_{xx}(k) \quad (3.55)$$

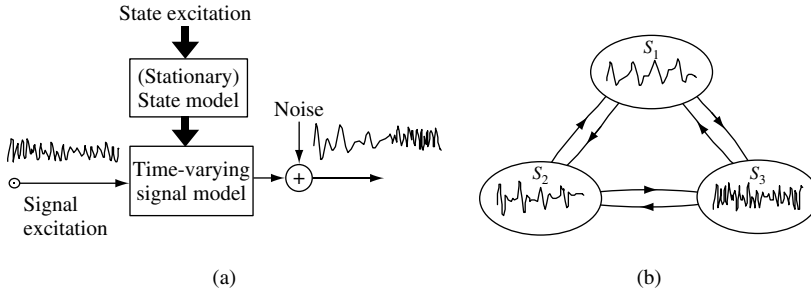


Figure 3.14 Two models for non-stationary processes: (a) a stationary process drives the parameters of a continuously time-varying model; (b) a finite-state model with each state having a different set of statistics.

From the definitions of strict-sense and wide-sense stationary processes, it is clear that a strict-sense stationary process is also wide-sense stationary, whereas the reverse is not necessarily true.

3.5.3 NONSTATIONARY PROCESSES

A random process is said to be a nonstationary process if its statistics vary with time. Most stochastic processes such as video and audio signals, financial data, meteorological data and biomedical signals are nonstationary, as they are generated by systems whose contents, environments and parameters vary or evolve over time. For example, speech is a nonstationary process generated by a time-varying articulatory system. The loudness and the frequency composition of speech change over time.

Time-varying processes may be modelled by some combination of stationary random models as illustrated in Figure 3.14. In Figure 3.14(a) a nonstationary process is modelled as the output of a time-varying system whose parameters are controlled by a stationary process. In Figure 3.14(b) a time-varying process is modelled by a Markov chain of time-invariant states, with each state having a different set of statistics or probability distributions. Finite-state statistical models for time-varying processes are discussed in detail in Chapter 5.

3.6 STATISTICS (EXPECTED VALUES) OF A RANDOM PROCESS

The expected values of a random process, also known as its statistics, are the mean, variance, correlation, power spectrum and the higher-order statistics of the process. Expected values play an indispensable role in signal processing. Furthermore, the probability models of a random process are usually expressed as functions of the expected values. For example, a Gaussian pdf is defined as an exponential function centred about the mean and with width and orientation determined by the covariance of the process, and a Poisson pdf is defined in terms of the mean of the process.

In signal processing applications, we often have a suitable statistical model of the process, e.g. a Gaussian pdf, and to complete the model we need the values of the expected parameters.

Furthermore, for many algorithms, such as noise reduction filters or linear prediction, what we essentially need is an estimate of the mean or the correlation function of the process.

The expected value of a function $h(X)$ of random process X , $h[X(m_1), X(m_2), \dots, X(m_M)]$, is defined as

$$\mathcal{E}\{h[X(m_1), \dots, X(m_M)]\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h(x_1, \dots, x_M) f_{X(m_1) \dots X(m_M)}(x_1, \dots, x_M) dx_1, \dots, dx_M \quad (3.56)$$

The most important, and widely used, expected values are the first-order moment, namely the mean value, and the second-order moments, namely the correlation, the covariance and the power spectrum.

3.6.1 THE MEAN VALUE

The mean value of a process plays an important part in signal processing and parameter estimation from noisy observations. For example, the optimal linear estimate of a signal from a noisy observation is a weighted interpolation between the mean value and the observed value of the noisy signal. The mean value of a vector process $[X(m_1), \dots, X(m_M)]$ is its average value across the space of the process, defined as

$$\mathcal{E}[X(m_1), \dots, X(m_M)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_1, \dots, x_M) f_{X(m_1) \dots X(m_M)}(x_1, \dots, x_M) dx_1, \dots, dx_M \quad (3.57)$$

For a segment of N samples of a signal $x(m)$, an estimate of the mean value is obtained as

$$\hat{\mu}_x = \frac{1}{N} \sum_{m=0}^{N-1} x(m) \quad (3.58)$$

Note that the estimate of the mean $\hat{\mu}_x$ in Equation (3.58), from a finite number of N samples, is itself a random variable with its own mean value, variance and probability distribution.

3.6.2 AUTOCORRELATION

The correlation function and its Fourier transform, the power spectral density, are extensively used in modelling and identification of patterns and structures in a signal process. Correlators play a central role in signal processing and telecommunication systems, including digital decoders, predictive coders, digital equalisers, delay estimators, classifiers and signal restoration systems.

The autocorrelation function of a random process $X(m)$, denoted by $r_{xx}(m_1, m_2)$, is defined as

$$\begin{aligned} r_{xx}(m_1, m_2) &= \mathcal{E}[x(m_1)x(m_2)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(m_1)x(m_2) f_{X(m_1), X(m_2)}[x(m_1), x(m_2)] dx(m_1) dx(m_2) \end{aligned} \quad (3.59)$$

The autocorrelation function $r_{xx}(m_1, m_2)$ is a measure of the similarity, or the mutual relation, of the outcomes of the process X at time instants m_1 and m_2 . If the outcome of a random process at time m_1 bears no relation to that at time m_2 , then $X(m_1)$ and $X(m_2)$ are said to be independent or uncorrelated and $r_{xx}(m_1, m_2) = 0$.

For a wide-sense stationary process, the autocorrelation function is time-invariant and depends on the time difference $m = m_1 - m_2$:

$$r_{xx}(m_1 + \tau, m_2 + \tau) = r_{xx}(m_1, m_2) = r_{xx}(m_1 - m_2) = r_{xx}(m) \quad (3.60)$$

The autocorrelation function of a real-valued wide-sense stationary process is a symmetric function with the following properties:

$$r_{xx}(-m) = r_{xx}(m) \quad (3.61)$$

$$r_{xx}(m) \leq r_{xx}(0) \quad (3.62)$$

For a segment of N samples of signal $x(m)$, the autocorrelation function is obtained as

$$r_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1-m} x(k)x(k+m) \quad (3.63)$$

Note that for a zero-mean signal, $r_{xx}(0)$ is the signal power. Autocorrelation of a signal can be obtained as the inverse Fourier transform of the magnitude spectrum as

$$r_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1} |X(f)|^2 e^{-j2\pi mk/N} \quad (3.64)$$

Example 3.14: Autocorrelation of a Periodic Signal – Estimation of Period

Autocorrelation can be used to calculate the repetition period T of a periodic signal such as the heart beat pulses shown in Figure 3.15(a). Figure 3.15(b) and (c) shows the estimate of the periods and the autocorrelation function, respectively, of the signal in Figure 3.15(a). Note that largest peak of the autocorrelation function occurs at a lag of zero at $r_{xx}(0)$ and the second largest peak occurs at a lag of T at $r_{xx}(T)$. Hence the difference of the time indices of the first and second peaks of the autocorrelation function provides an estimate of the period of a signal.

Example 3.15

Autocorrelation of the output of a linear time-invariant (LTI) system. Let $x(m)$, $y(m)$ and $h(m)$ denote the input, the output and the impulse response of an LTI system, respectively. The input–output relation is given by

$$y(m) = \sum_i h(i)x(m-i) \quad (3.65)$$

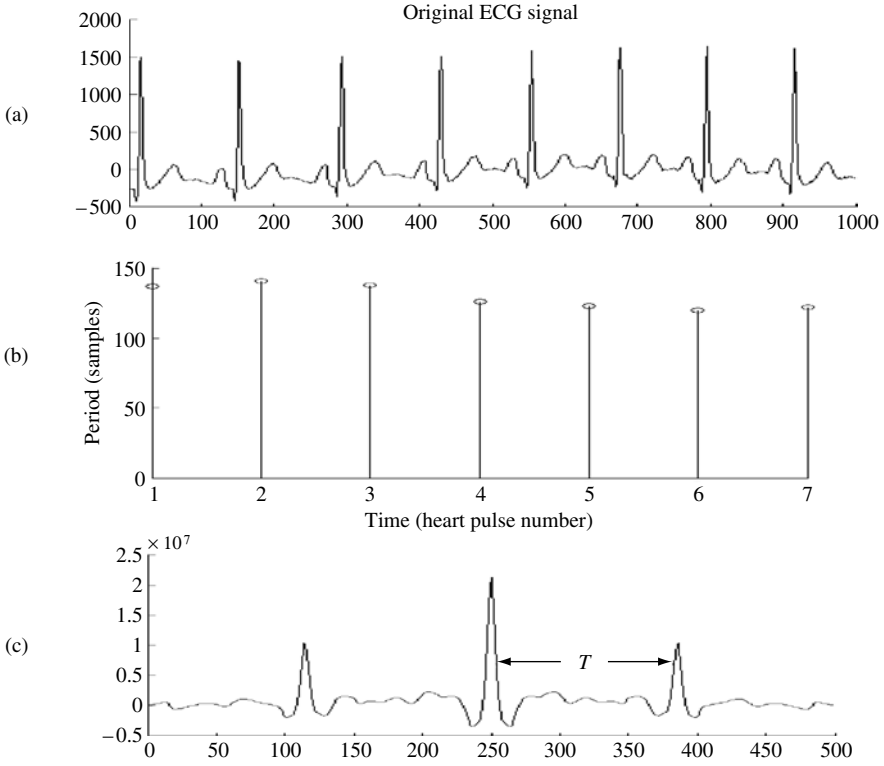


Figure 3.15 (a) Heart beat signal, electrocardiograph (ECG); (b) variation of period with time; (c) autocorrelation function of ECG.

The autocorrelation function of the output signal $y(m)$ can be related to the autocorrelation of the input signal, $x(m)$, by

$$\begin{aligned}
 r_{yy}(k) &= \mathcal{E}[y(m)y(m+k)] \\
 &= \sum_i \sum_j h(i)h(j)\mathcal{E}[x(m-i)x(m+k-j)] \\
 &= \sum_i \sum_j h(i)h(j)r_{xx}(k+i-j)
 \end{aligned}
 \tag{3.66}$$

When the input $x(m)$ is an uncorrelated zero-mean random signal with a unit variance, its autocorrelation is given by

$$r_{xx}(l) = \begin{cases} 1 & l = 0 \\ 0 & l \neq 0 \end{cases}
 \tag{3.67}$$

then $r_{xx}(k+i-j) = 1$ when $j = k+i$ and Equation (3.66) becomes

$$r_{yy}(k) = \sum_i h(i)h(k+i)
 \tag{3.68}$$

3.6.3 AUTOCOVARIANCE

The autocovariance function $c_{xx}(m_1, m_2)$ of a random process $X(m)$ is a measure of the scatter, or the dispersion, of the process about the mean value, and is defined as

$$\begin{aligned} c_{xx}(m_1, m_2) &= \mathcal{E}\{[x(m_1) - \mu_x(m_1)][x(m_2) - \mu_x(m_2)]\} \\ &= r_{xx}(m_1, m_2) - \mu_x(m_1)\mu_x(m_2) \end{aligned} \quad (3.69)$$

where $\mu_x(m)$ is the mean of $X(m)$. Note that for a zero-mean process, the autocorrelation and the autocovariance functions are identical. Note also that $c_{xx}(m_1, m_1)$ is the variance of the process. For a stationary process the autocovariance function of Equation (3.69) becomes

$$c_{xx}(m_1, m_2) = c_{xx}(m_1 - m_2) = r_{xx}(m_1 - m_2) - \mu_x^2 \quad (3.70)$$

3.6.4 POWER SPECTRAL DENSITY

The power spectral density (PSD) function, also called the power spectrum, of a process gives the spectrum of the distribution of power at different frequencies of vibrations along the frequency axis. It can be shown that the power spectrum of a wide-sense stationary process, $X(m)$, is the Fourier transform of the autocorrelation function:

$$\begin{aligned} P_{XX}(f) &= \mathcal{E}[X(f)X^*(f)] \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(k) e^{-j2\pi f m} \end{aligned} \quad (3.71)$$

where $r_{xx}(m)$ and $P_{XX}(f)$ are the autocorrelation and power spectrum of $x(m)$, respectively, and f is the frequency variable. For a real-valued stationary process, the autocorrelation is symmetric, and the power spectrum may be written as

$$P_{XX}(f) = r_{xx}(0) + \sum_{m=1}^{\infty} 2r_{xx}(m) \cos(2\pi f m) \quad (3.72)$$

The power spectral density is a real-valued non-negative function, expressed in units of watts per hertz. From Equation (3.71), the autocorrelation sequence of a random process may be obtained as the inverse Fourier transform of the power spectrum as

$$r_{xx}(m) = \int_{-1/2}^{1/2} P_{XX}(f) e^{j2\pi f m} df \quad (3.73)$$

Note that the autocorrelation and the power spectrum represent the second-order statistics of a process in time and frequency domains, respectively.

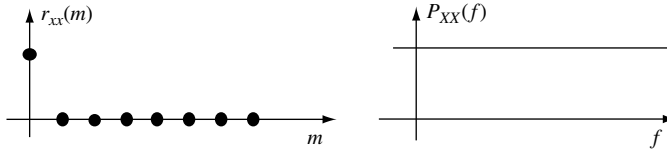


Figure 3.16 Autocorrelation and power spectrum of white noise.

Example 3.16

Power spectrum and autocorrelation of white noise (Figure 3.16). A noise process with uncorrelated independent samples is called a ‘white noise process’. The autocorrelation of a stationary white noise $n(m)$ is defined as:

$$r_{nn}(k) = \mathcal{E}[n(m)n(m+k)] = \begin{cases} \text{Noise power} & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (3.74)$$

Equation (3.74) is a mathematical statement of the definition of an uncorrelated white noise process. The equivalent description in the frequency domain is derived by taking the Fourier transform of $r_{nn}(k)$:

$$P_{NN}(f) = \sum_{k=-\infty}^{\infty} r_{nn}(k)e^{-j2\pi fk} = r_{nn}(0) = \text{noise power} \quad (3.75)$$

From Equation (3.75), the power spectrum of a stationary white noise process is spread equally across all time instances and across all frequency bins. White noise is one of the most difficult types of noise to remove, because it does not have a localised structure in either the time domain or the frequency domain.

Example 3.17

Power spectrum and autocorrelation of a discrete-time impulse. The autocorrelation of a discrete-time impulse with amplitude A , $A\delta(m)$, is defined as:

$$r_{\delta\delta}(k) = \mathcal{E}[A^2\delta(m)\delta(m+k)] = \begin{cases} A^2 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (3.76)$$

The power spectrum of the impulse is the obtained by taking the Fourier transform of $r_{\delta\delta}(k)$ as:

$$P_{\Delta\Delta}(f) = \sum_{k=-\infty}^{\infty} r_{\delta\delta}(k)e^{-j2\pi fk} = A^2 \quad (3.77)$$

Example 3.18

Autocorrelation and power spectrum of impulsive noise. Impulsive noise is a random, binary-state (‘on/off’) sequence of impulses of random amplitudes and random time of occurrence.

A random impulsive noise sequence, $n_i(m)$, can be modelled as an amplitude-modulated random binary sequence as

$$n_i(m) = n(m)b(m) \quad (3.78)$$

where $b(m)$ is a binary-state random sequence that indicates the presence or absence of an impulse, and $n(m)$ is a random noise process. Assuming that impulsive noise is an uncorrelated process, the autocorrelation of impulsive noise can be defined as a binary-state process as

$$r_{nn}(k, m) = \mathcal{E}[n_i(m)n_i(m+k)] = \sigma_n^2 \delta(k)b(m) \quad (3.79)$$

where σ_n^2 is the noise variance. Note that, in Equation (3.79), the autocorrelation is expressed as a binary-state function that depends on the on/off state of impulsive noise at time m . The power spectrum of an impulsive noise sequence is obtained by taking the Fourier transform of the autocorrelation function as

$$P_{NN}(f, m) = \sigma_n^2 b(m) \quad (3.80)$$

3.6.5 JOINT STATISTICAL AVERAGES OF TWO RANDOM PROCESSES

In many signal processing problems, for example in processing the outputs of an array of sensors, we have more than one random process. Joint statistics and joint distributions are used to describe the statistical inter-relationship between two or more random processes. For two discrete-time random processes, $x(m)$ and $y(m)$, the joint pdf is denoted by

$$f_{X(m_1)\dots X(m_M), Y(n_1)\dots Y(n_N)}(x_1, \dots, x_M, y_1, \dots, y_N) \quad (3.81)$$

When two random processes, $X(m)$ and $Y(m)$, are uncorrelated, the joint pdf can be expressed as product of the pdfs of each process as

$$\begin{aligned} & f_{X(m_1)\dots X(m_M), Y(n_1)\dots Y(n_N)}(x_1, \dots, x_M, y_1, \dots, y_N) \\ &= f_{X(m_1)\dots X(m_M)}(x_1, \dots, x_M) f_{Y(n_1)\dots Y(n_N)}(y_1, \dots, y_N) \end{aligned} \quad (3.82)$$

3.6.6 CROSS-CORRELATION AND CROSS-COVARIANCE

The cross-correlation of two random process $x(m)$ and $y(m)$ is defined as

$$\begin{aligned} r_{xy}(m_1, m_2) &= \mathcal{E}[x(m_1)y(m_2)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(m_1)y(m_2) f_{X(m_1)Y(m_2)}[x(m_1), y(m_2)] dx(m_1) dy(m_2) \end{aligned} \quad (3.83)$$

For wide-sense stationary processes, the cross-correlation function $r_{xy}(m_1, m_2)$ depends only on the time difference $m = m_1 - m_2$:

$$r_{xy}(m_1 + \tau, m_2 + \tau) = r_{xy}(m_1, m_2) = r_{xy}(m_1 - m_2) = r_{xy}(m) \quad (3.84)$$

The cross-covariance function is defined as

$$\begin{aligned} c_{xy}(m_1, m_2) &= \mathcal{E}\{[x(m_1) - \mu_x(m_1)][y(m_2) - \mu_y(m_2)]\} \\ &= r_{xy}(m_1, m_2) - \mu_x(m_1)\mu_y(m_2) \end{aligned} \quad (3.85)$$

Note that for zero-mean processes, the cross-correlation and the cross-covariance functions are identical. For a wide-sense stationary process the cross-covariance function of Equation (3.85) becomes

$$c_{xy}(m_1, m_2) = c_{xy}(m_1 - m_2) = r_{xy}(m_1 - m_2) - \mu_x\mu_y \quad (3.86)$$

Example 3.19: Time-delay Estimation

Consider two signals, $y_1(m)$ and $y_2(m)$, each composed of an information bearing signal, $x(m)$, and an additive noise, given by

$$y_1(m) = x(m) + n_1(m) \quad (3.87)$$

$$y_2(m) = Ax(m - D) + n_2(m) \quad (3.88)$$

where A is an amplitude factor and D is a time delay variable. The cross-correlation of the signals, $y_1(m)$ and $y_2(m)$, yields

$$\begin{aligned} r_{y_1y_2}(k) &= \mathcal{E}[y_1(m)y_2(m+k)] \\ &= \mathcal{E}\{[x(m) + n_1(m)][Ax(m - D + k) + n_2(m+k)]\} \\ &= Ar_{xx}(k - D) + r_{xn_2}(k) + Ar_{xn_1}(k - D) + r_{n_1n_2}(k) \end{aligned} \quad (3.89)$$

Assuming that the signal and noise are uncorrelated, we have $r_{y_1y_2}(k) = Ar_{xx}(k - D)$. As shown in Figure 3.17, the cross-correlation function has its maximum at the lag D .

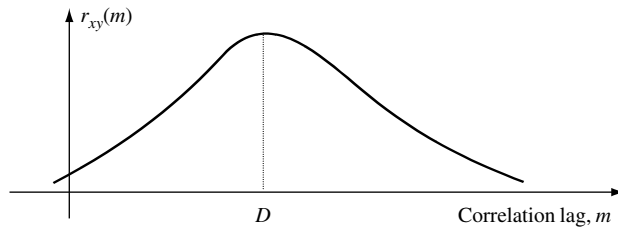


Figure 3.17 The peak of the cross-correlation of two delayed signals can be used to estimate the time delay, D .

3.6.7 CROSS-POWER SPECTRAL DENSITY AND COHERENCE

The cross-power spectral density of two random processes, $X(m)$ and $Y(m)$, is defined as the Fourier transform of their cross-correlation function:

$$\begin{aligned} P_{XY}(f) &= \mathcal{E}[X(f)Y^*(f)] \\ &= \sum_{m=-\infty}^{\infty} r_{xy}(m)e^{-j2\pi fm} \end{aligned} \quad (3.90)$$

Cross-power spectral density of two processes is a measure of the similarity, or coherence, of their power spectra. The coherence, or spectral coherence, of two random processes is a normalised form of the cross-power spectral density, defined as

$$C_{XY}(f) = \frac{P_{XY}(f)}{\sqrt{P_{XX}(f)P_{YY}(f)}} \quad (3.91)$$

The coherence function is used in applications such as time-delay estimation and signal-to-noise ratio measurements.

3.6.8 ERGODIC PROCESSES AND TIME-AVERAGED STATISTICS

In many signal processing problems, there is only a single realisation of a random process from which its statistical parameters, such as the mean, the correlation and the power spectrum, can be estimated. In these cases, time-averaged statistics, obtained from averages along the time dimension of a single realisation of the process, are used instead of the ensemble averages obtained across the space of different realisations of the process. This section considers ergodic random processes for which time-averages can be used instead of ensemble averages.

A stationary stochastic process is said to be ergodic if it exhibits the same statistical characteristics along the time dimension of a single realisation as across the space (or ensemble) of different realisations of the process. Over a very long time, a single realisation of an ergodic process takes on all the values, the characteristics and the configurations exhibited across the entire space of the process. For an ergodic process $\{x(m, s)\}$, we have

$$\text{Statistical averages}_{\text{alongtime } m} [x(m, s)] = \text{Statistical averages}_{\text{acrossspace } s} [x(m, s)] \quad (3.92)$$

where the *statistical averages* $[\cdot]$ function refers to any statistical operation such as the mean, the variance, the power spectrum, etc.

3.6.9 MEAN-ERGODIC PROCESSES

The time-averaged estimate of the mean of a signal $x(m)$ obtained from N samples is given by

$$\hat{\mu}_X = \frac{1}{N} \sum_{m=0}^{N-1} x(m) \quad (3.93)$$

A stationary process is mean-ergodic if the time-averaged value of an infinitely long realisation of the process is the same as the ensemble-mean taken across the space of the process. Therefore, for a mean-ergodic process, we have

$$\lim_{N \rightarrow \infty} \mathcal{E}[\hat{\mu}_X] = \mu_X \quad (3.94)$$

$$\lim_{N \rightarrow \infty} \text{var}[\hat{\mu}_X] = 0 \quad (3.95)$$

where μ_X is the ensemble average of the process. The time-averaged estimate of the mean of a signal, obtained from a random realisation of the process, is itself a random variable, with its own mean, variance and probability density function. If the number of observation samples, N , is relatively large, then, from the central limit theorem, the probability density function of the estimate $\hat{\mu}_X$ is Gaussian. The expectation of $\hat{\mu}_X$ is given by

$$\mathcal{E}[\hat{\mu}_X] = \mathcal{E} \left[\frac{1}{N} \sum_{m=0}^{N-1} x(m) \right] = \frac{1}{N} \sum_{m=0}^{N-1} \mathcal{E}[x(m)] = \frac{1}{N} \sum_{m=0}^{N-1} \mu_X = \mu_X \quad (3.96)$$

From Equation (3.96), the time-averaged estimate of the mean is unbiased. The variance of $\hat{\mu}_X$ is given by

$$\begin{aligned} \text{Var}[\hat{\mu}_X] &= \mathcal{E}[\hat{\mu}_X^2] - \mathcal{E}^2[\hat{\mu}_X] \\ &= \mathcal{E}[\hat{\mu}_X^2] - \mu_X^2 \end{aligned} \quad (3.97)$$

Now the term $\mathcal{E}[\hat{\mu}_X^2]$ in Equation (3.97) may be expressed as

$$\begin{aligned} \mathcal{E}[\hat{\mu}_X^2] &= \mathcal{E} \left\{ \left[\frac{1}{N} \sum_{m=0}^{N-1} x(m) \right] \left[\frac{1}{N} \sum_{k=0}^{N-1} x(k) \right] \right\} \\ &= \frac{1}{N} \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N} \right) r_{xx}(m) \end{aligned} \quad (3.98)$$

Substitution of Equation (3.98) in Equation (3.97) yields

$$\begin{aligned} \text{Var}[\hat{\mu}_X^2] &= \frac{1}{N} \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N} \right) r_{xx}(m) - \mu_X^2 \\ &= \frac{1}{N} \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N} \right) c_{xx}(m) \end{aligned} \quad (3.99)$$

The condition for a process to be mean-ergodic in the mean square error sense is

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N} \right) c_{xx}(m) = 0 \quad (3.100)$$

3.6.10 CORRELATION-ERGODIC PROCESSES

The time-averaged estimate of the autocorrelation of a random process, estimated from a segment of N samples, is given by

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1} x(k)x(k+m) \quad (3.101)$$

The estimate of autocorrelation $\hat{r}_{xx}(m)$ is itself a random variable with its own mean, variance and probability distribution. A process is correlation-ergodic, in the mean square error sense, if

$$\lim_{N \rightarrow \infty} \mathcal{E}[\hat{r}_{xx}(m)] = r_{xx}(m) \quad (3.102)$$

$$\lim_{N \rightarrow \infty} \text{Var}[\hat{r}_{xx}(m)] = 0 \quad (3.103)$$

where $r_{xx}(m)$ is the ensemble-averaged autocorrelation. Taking the expectation of $\hat{r}_{xx}(m)$ shows that it is an unbiased estimate, since

$$\mathcal{E}[\hat{r}_{xx}(m)] = \mathcal{E}\left[\frac{1}{N} \sum_{k=0}^{N-1} x(k)x(k+m)\right] = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{E}[x(k)x(k+m)] = r_{xx}(m) \quad (3.104)$$

The variance of $\hat{r}_{xx}(m)$ is given by

$$\text{Var}[\hat{r}_{xx}(m)] = \mathcal{E}[\hat{r}_{xx}^2(m)] - r_{xx}^2(m) \quad (3.105)$$

The term $\mathcal{E}[\hat{r}_{xx}^2(m)]$ in Equation (3.105) may be expressed as

$$\begin{aligned} \mathcal{E}[\hat{r}_{xx}^2(m)] &= \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{E}[x(k)x(k+m)x(j)x(j+m)] \\ &= \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{E}[z(k, m)z(j, m)] \\ &= \frac{1}{N} \sum_{k=-N+1}^{N-1} \left(1 - \frac{|k|}{N}\right) r_{zz}(k, m) \end{aligned} \quad (3.106)$$

where $z(i, m) = x(i)x(i+m)$. The condition for correlation ergodicity, in the mean square error sense, is given by

$$\lim_{N \rightarrow \infty} \left[\frac{1}{N} \sum_{k=-N+1}^{N-1} \left(1 - \frac{|k|}{N}\right) r_{zz}(k, m) - r_{xx}^2(m) \right] = 0 \quad (3.107)$$

3.7 SOME USEFUL CLASSES OF RANDOM PROCESSES

In this section, we consider some important classes of random processes that are extensively used in communication signal processing for such applications as decoding, channel equalisation, modelling of noise and fading and pattern recognition.

3.7.1 GAUSSIAN (NORMAL) PROCESS

The Gaussian process, also called the normal process, is the most widely applied of all probability models. Some advantages of Gaussian probability models are the following:

- (1) Gaussian pdfs can model the distribution of many processes, including some important classes of signals and noise;
- (2) non-Gaussian processes can be approximated by a weighted combination (i.e. a mixture) of a number of Gaussian pdfs of appropriate means and variances;
- (3) optimal estimation methods based on Gaussian models often result in linear and mathematically tractable solutions;
- (4) the sum of many independent random processes has a Gaussian distribution; this is known as the *central limit theorem*.

A scalar-valued Gaussian random variable is described by the following probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(x - \mu_x)^2}{2\sigma_x^2}\right] \quad (3.108)$$

where μ_x and σ_x^2 are the mean and the variance of the random variable x . Note that the argument of the exponential of a Gaussian function, $(x - \mu_x)^2/2\sigma_x^2$, is a variance-normalised distance. The Gaussian process of Equation (3.108) is also denoted by $\mathcal{N}(x, \mu_x, \sigma_x^2)$. The maximum of a Gaussian pdf occurs at the mean μ_x , and is given by

$$f_X(\mu_x) = \frac{1}{\sqrt{2\pi}\sigma_x} \quad (3.109)$$

From Equation (3.108), the Gaussian pdf of x decreases exponentially with the distance x from the mean value μ_x . The cumulative distribution function $F(x)$ is given by

$$F_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^x \exp\left(-\frac{(\chi - \mu_x)^2}{2\sigma_x^2}\right) d\chi \quad (3.110)$$

Figure 3.18 shows the bell-shaped pdf and the cdf of a Gaussian model. The most probable values of a Gaussian process happen around the mean, and the probability of a value decreases exponentially with increasing distance from the mean value. The total area under the pdf curve is 1. Note that the area under the pdf curve one standard deviation on each side of the mean value ($\mu \pm \sigma$) is 0.682, the area two standard deviations on each side the mean value ($\mu \pm 2\sigma$) is 0.955 and the area three standard deviations on each side the mean value ($\mu \pm 3\sigma$) is 0.997.

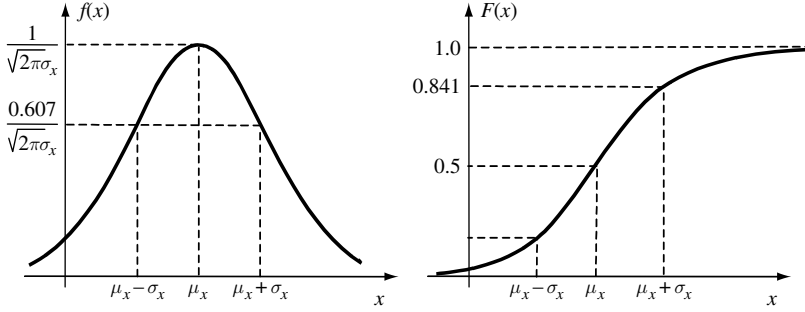


Figure 3.18 Gaussian probability density and cumulative density functions.

3.7.2 MULTIVARIATE GAUSSIAN PROCESS

Multivariate densities model vector-valued processes. Consider a P -variate Gaussian vector process, $\mathbf{x} = [x(m_0), x(m_1), \dots, x(m_{P-1})]^T$, with mean vector $\boldsymbol{\mu}_x$, and covariance matrix $\boldsymbol{\Sigma}_{xx}$. The multivariate Gaussian pdf of \mathbf{x} is given by

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_{xx}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_x)^T \boldsymbol{\Sigma}_{xx}^{-1} (\mathbf{x} - \boldsymbol{\mu}_x) \right] \quad (3.111)$$

where the mean vector $\boldsymbol{\mu}_x$ is defined as

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mathcal{E}[x(m_0)] \\ \mathcal{E}[x(m_1)] \\ \vdots \\ \mathcal{E}[x(m_{P-1})] \end{bmatrix} \quad (3.112)$$

and the covariance matrix $\boldsymbol{\Sigma}_{xx}$ is given by

$$\boldsymbol{\Sigma}_{xx} = \begin{bmatrix} c_{xx}(m_0, m_0) & c_{xx}(m_0, m_1) & \cdots & c_{xx}(m_0, m_{P-1}) \\ c_{xx}(m_1, m_0) & c_{xx}(m_1, m_1) & \cdots & c_{xx}(m_1, m_{P-1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}(m_{P-1}, m_0) & c_{xx}(m_{P-1}, m_1) & \cdots & c_{xx}(m_{P-1}, m_{P-1}) \end{bmatrix} \quad (3.113)$$

The Gaussian process of Equation (3.111) is also denoted by $\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$. If the elements of a vector process are uncorrelated then the covariance matrix is a diagonal matrix with zeros in the off-diagonal elements. In this case the multivariate pdf may be described as the product of the pdfs of the individual elements of the vector:

$$f_X \{ \mathbf{x} = [x(m_0), \dots, x(m_{P-1})]^T \} = \prod_{i=0}^{P-1} \frac{1}{\sqrt{2\pi}\sigma_{x_i}} \exp \left\{ -\frac{[x(m_i) - \mu_{x_i}]^2}{2\sigma_{x_i}^2} \right\} \quad (3.114)$$

Example 3.20

Conditional multivariate Gaussian probability density function. Consider two vector realisations $\mathbf{x}(m)$ and $\mathbf{y}(m)$ from two vector-valued correlated stationary Gaussian processes, $\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$ and $\mathcal{N}(\mathbf{y}, \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_{yy})$. The joint probability density function of $\mathbf{x}(m)$ and $\mathbf{y}(m)$ is a multivariate Gaussian density, $\mathcal{N}\{\mathbf{x}(m), \mathbf{y}(m), \boldsymbol{\mu}_{(x,y)}, \boldsymbol{\Sigma}_{(x,y)}\}$, with mean vector and covariance matrix given by

$$\boldsymbol{\mu}_{(x,y)} = \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix} \quad (3.115)$$

$$\boldsymbol{\Sigma}_{(x,y)} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \quad (3.116)$$

The conditional density of $\mathbf{x}(m)$ given $\mathbf{y}(m)$ is given from Bayes' rule as

$$f_{(x|y)}[\mathbf{x}(m) | \mathbf{y}(m)] = \frac{f_{X,Y}[\mathbf{x}(m), \mathbf{y}(m)]}{f_Y[\mathbf{y}(m)]} \quad (3.117)$$

It can be shown that the conditional density is also a multivariate Gaussian with its mean vector and covariance matrix given by

$$\begin{aligned} \boldsymbol{\mu}_{(x|y)} &= \mathcal{E}[\mathbf{x}(m) | \mathbf{y}(m)] \\ &= \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} (\mathbf{y} - \boldsymbol{\mu}_y) \end{aligned} \quad (3.118)$$

$$\boldsymbol{\Sigma}_{(x|y)} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx} \quad (3.119)$$

3.7.3 MIXTURE GAUSSIAN PROCESS

The probability density functions of many random processes, such as speech, are non-Gaussian. A non-Gaussian pdf may be approximated by a weighted sum (i.e. a mixture) of a number of Gaussian densities of appropriate mean vectors and covariance matrices. A mixture Gaussian density with M components is defined as

$$f_X(\mathbf{x}) = \sum_{i=1}^M P_i \mathcal{N}_i(\mathbf{x}, \boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_{xx_i}) \quad (3.120)$$

where $\mathcal{N}_i(\mathbf{x}, \boldsymbol{\mu}_{x_i}, \boldsymbol{\Sigma}_{xx_i})$ is a multivariate Gaussian density with mean vector $\boldsymbol{\mu}_{x_i}$ and covariance matrix $\boldsymbol{\Sigma}_{xx_i}$, and P_i are the mixing coefficients. The parameter P_i is the prior probability of the i th component of the mixture, given by

$$P_i = \frac{N_i}{\sum_{j=1}^M N_j} \quad (3.121)$$

where N_i is the number of observations, of the process, associated with the mixture i . Figure 3.19 shows a non-Gaussian pdf modelled as a mixture of five Gaussian pdfs. Algorithms developed for Gaussian processes can be extended to mixture Gaussian densities.

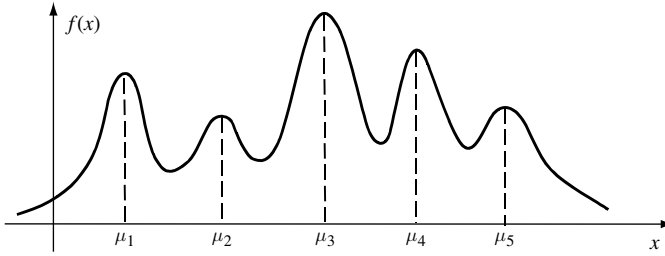


Figure 3.19 A Gaussian mixture model (GMM) pdf.

3.7.4 A BINARY-STATE GAUSSIAN PROCESS

A simple example of a binary-state process is the observations at the output of communication system with the input signal consisting of a binary sequence ('0' and '1') process.

Consider a random process, $x(m)$, with two states, s_0 and s_1 , such that in the state s_0 the process has a Gaussian pdf with mean $\mu_{x,0}$ and variance $\sigma_{x,0}^2$, and in the state s_1 the process is also Gaussian with mean $\mu_{x,1}$ and variance $\sigma_{x,0}^2$. The state-dependent pdf of $x(m)$ can be expressed as

$$f_{X|S}[x(m)|s_i] = \frac{1}{\sqrt{2\pi}\sigma_{x,i}} \exp\left\{-\frac{1}{2\sigma_{x,i}^2}[x(m) - \mu_{x,i}]^2\right\}, \quad i = 0, 1 \quad (3.122)$$

The joint probability distribution of the binary-valued state s_i and the continuous-valued signal $x(m)$ can be expressed as

$$\begin{aligned} f_{X,S}[x(m), s_i] &= f_{X|S}[x(m)|s_i]P_S(s_i) \\ &= \frac{1}{\sqrt{2\pi}\sigma_{x,i}} \exp\left\{-\frac{1}{2\sigma_{x,i}^2}[x(m) - \mu_{x,i}]^2\right\}P_S(s_i) \end{aligned} \quad (3.123)$$

where $P_S(s_i)$ is the state probability. For a multistate process we have the following probabilistic relations between the joint and marginal probabilities:

$$\sum_S f_{X,S}[x(m), s_i] = f_X[x(m)] \quad (3.124)$$

$$\int_X f_{X,S}[x(m), s_i] dx = P_S(s_i) \quad (3.125)$$

and

$$\sum_S \int_X f_{X,S}[x(m), s_i] dx = 1 \quad (3.126)$$

Note that, in a multistate model, the statistical parameters of the process *switch* between a number of different states, whereas in a single-state mixture pdf, a *weighted* combination of a number of pdfs models the process. In Chapter 5 on hidden Markov models we consider multistate models with a mixture Gaussian pdf per state.

3.7.5 POISSON PROCESS

The Poisson process is a continuous-time integer-valued counting process, used for modelling the occurrences of a random discrete event in various time intervals. An important area of application of the Poisson process is in the queuing theory for the analysis and modelling of the distributions of demand on a service facility such as a telephone network, a computer network, a financial service, a transport network, a petrol station, etc. Other applications of the Poisson distribution include the counting of the number of particles emitted in particle physics, the number of times that a component may fail in a system, and the modelling of radar clutter, shot noise and impulsive noise.

Consider an event-counting process, $X(t)$, in which the probability of occurrence of the event is governed by a rate function $\lambda(t)$, such that the probability that an event occurs in a small time interval Δt is

$$\text{Prob}[1 \text{ occurrence in the interval } (t, t + \Delta t)] = \lambda(t)\Delta t \quad (3.127)$$

Assuming that, in the small interval Δt , no more than one occurrence of the event is possible, the probability of no occurrence of the event in a time interval Δt is given by

$$\text{Prob}[0 \text{ occurrence in the interval } (t, t + \Delta t)] = 1 - \lambda(t)\Delta t \quad (3.128)$$

When the parameter $\lambda(t)$ is independent of time, $\lambda(t) = \lambda$, the process is called a 'homogeneous Poisson process'. Now, for a homogeneous Poisson process, consider the probability of k occurrences of an event in a time interval $t + \Delta t$, denoted by $P[k, (0, t + \Delta t)]$:

$$\begin{aligned} P[k, (0, t + \Delta t)] &= P[k, (0, t)]P[0, (t, t + \Delta t)] + P[k - 1, (0, t)]P[1, (t, t + \Delta t)] \\ &= P[k, (0, t)](1 - \lambda\Delta t) + P[k - 1, (0, t)]\lambda\Delta t \end{aligned} \quad (3.129)$$

Rearranging Equation (3.129), and letting Δt tend to zero, we obtain the following linear differential equation:

$$\lim_{\Delta t \rightarrow 0} \frac{P[k(0, t + \Delta t)] - P[k(0, t)]}{\Delta t} = \frac{dP(k, t)}{dt} = -\lambda P(k, t) + \lambda P(k - 1, t) \quad (3.130)$$

where $P(k, t) = P[k, (0, t)]$. The solution of this differential equation is given by

$$P(k, t) = \lambda e^{-\lambda t} \int_0^t P(k - 1, \tau) e^{\lambda \tau} d\tau \quad (3.131)$$

Equation (3.131) can be solved recursively: starting with $P(0, t) = e^{-\lambda t}$ and $P(1, t) = \lambda t e^{-\lambda t}$, we obtain the Poisson density

$$P(k, t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad (3.132)$$

From Equation (3.132), it is easy to show that, for a homogenous Poisson process, the probability of k occurrences of an event in a time interval (t_1, t_2) is given by

$$P[k, (t_1, t_2)] = \frac{[\lambda(t_2 - t_1)]^k}{k!} e^{-\lambda(t_2 - t_1)} \quad (3.133)$$

A Poisson counting process $X(t)$ is incremented by one every time the event occurs. From Equation (3.132), the mean and variance of a Poisson counting process $X(t)$ are

$$\mathcal{E}[X(t)] = \lambda t \quad (3.134)$$

$$r_{XX}(t_1, t_2) = \mathcal{E}[X(t_1)X(t_2)] = \lambda^2 t_1 t_2 + \lambda \min(t_1, t_2) \quad (3.135)$$

$$\text{Var}[X(t)] = \mathcal{E}[X^2(t)] - \mathcal{E}^2[X(t)] = \lambda t \quad (3.136)$$

Note that the variance of a Poisson process is equal to its mean value.

3.7.6 SHOT NOISE

Shot noise results from randomness in directional flow of particles, for example in the flow of electrons from the cathode to the anode of a cathode ray tube, the flow of photons in a laser beam, the flow and recombination of electrons and holes in semiconductors, and the flow of photoelectrons emitted in photodiodes. Shot noise has the form of a random pulse sequence that may be modelled as the response of a linear filter excited by a Poisson-distributed binary impulse sequence (Figure 3.20).

Consider a Poisson-distributed binary-valued impulse process, $x(t)$. Divide the time axis into uniform short intervals, Δt , such that only one occurrence of an impulse is possible within each time interval. Let $x(m\Delta t)$ be '1' if an impulse is present in the interval $m\Delta t$ to $(m+1)\Delta t$, and '0' otherwise. For $x(m\Delta t)$, we obtain the mean and correlation functions as

$$\mathcal{E}[x(m\Delta t)] = 1 \times P[x(m\Delta t) = 1] + 0 \times P[x(m\Delta t) = 0] = \lambda \Delta t \quad (3.137)$$

and

$$\mathcal{E}[x(m\Delta t)x(n\Delta t)] = \begin{cases} 1 \times P[x(m\Delta t) = 1] = \lambda \Delta t, & m = n \\ 1 \times P[x(m\Delta t) = 1] \times P[x(n\Delta t) = 1] = (\lambda \Delta t)^2, & m \neq n \end{cases} \quad (3.138)$$

A shot noise process $y(m)$ is modelled as the output of a linear system with an impulse response $h(t)$, excited by a Poisson-distributed binary impulse input $x(t)$:

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau) d\tau \\ &= \sum_{k=-\infty}^{\infty} x(m\Delta t)h(t-m\Delta t) \end{aligned} \quad (3.139)$$

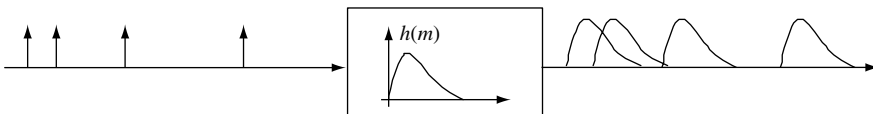


Figure 3.20 Shot noise is modelled as the output of a filter excited with a process.

where the binary signal $x(m\Delta t)$ can assume a value of 0 or 1. In Equation (3.139) it is assumed that the impulses happen at the beginning of each interval. This assumption becomes more valid as Δt becomes smaller. The expectation of $y(t)$ is obtained as

$$\begin{aligned}\mathcal{E}[y(t)] &= \sum_{k=-\infty}^{\infty} \mathcal{E}[x(m\Delta t)] h(t - m\Delta t) \\ &= \sum_{k=-\infty}^{\infty} \lambda \Delta t h(t - m\Delta t)\end{aligned}\quad (3.140)$$

and

$$\begin{aligned}r_{yy}(t_1, t_2) &= \mathcal{E}[y(t_1)y(t_2)] \\ &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \mathcal{E}[x(m\Delta t)x(n\Delta t)] h(t_1 - m\Delta t)h(t_2 - m\Delta t)\end{aligned}\quad (3.141)$$

Using Equation (3.138) in Equation (3.141), the autocorrelation of $y(t)$ can be obtained as

$$\begin{aligned}r_{yy}(t_1, t_2) &= \sum_{n=-\infty}^{\infty} (\lambda \Delta t) h(t_1 - m\Delta t)h(t_2 - m\Delta t) \\ &\quad + \sum_{m=-\infty}^{\infty} \sum_{\substack{n=-\infty \\ n \neq m}}^{\infty} (\lambda \Delta t)^2 h(t_1 - m\Delta t)h(t_2 - n\Delta t)\end{aligned}\quad (3.142)$$

3.7.7 POISSON–GAUSSIAN MODEL FOR CLUTTERS AND IMPULSIVE NOISE

An impulsive noise process consists of a sequence of short-duration pulses of random amplitude and random time of occurrence whose shape and duration depends on the characteristics of the channel through which the impulse propagates. A Poisson process can be used to model the random time of occurrence of impulsive noise, and a Gaussian process can model the random amplitude of the impulses. Finally, the finite duration character of real impulsive noise may be modelled by the impulse response of linear filter. The Poisson–Gaussian impulsive noise model is given by

$$x(m) = \sum_{k=-\infty}^{\infty} A_k h(m - \tau_k) \quad (3.143)$$

where $h(m)$ is the response of a linear filter that models the shape of impulsive noise, A_k is a zero-mean Gaussian process of variance σ_x^2 and τ_k denotes the instances of occurrences of impulses modelled by a Poisson process. The output of a filter excited by a Poisson-distributed sequence of Gaussian amplitude impulses can also be used to model clutters in radar. Clutters are due to reflection of radar pulses from a multitude of background surfaces and objects other than the intended radar target.

3.7.8 MARKOV PROCESSES

Markov processes are used to model the trajectory of a random process and to describe the dependency of the outcome of a process at any given time on the past outcomes of the process. Applications of Markov models include modelling the trajectory of a process in signal estimation and pattern recognition for speech, image and biomedical signal processing.

A first-order discrete-time Markov process is defined as one in which the state or the value of the process at time m depends only on its state or value at time $m - 1$ and is independent of the states or values of the process before time $m - 1$. In probabilistic terms, a first-order Markov process can be defined as

$$\begin{aligned} f_X[x(m) = x_m | x(m-1) = x_{m-1}, \dots, x(m-N) = x_{m-N}] \\ = f_X[x(m) = x_m | x(m-1) = x_{m-1}] \end{aligned} \quad (3.144)$$

The marginal density of a Markov process at time m can be obtained by integrating the conditional density over all values of $x(m-1)$:

$$f_X[x(m) = x_m] = \int_{-\infty}^{\infty} f_X[x(m) = x_m | x(m-1) = x_{m-1}] f_X[x(m-1) = x_{m-1}] dx_{m-1} \quad (3.145)$$

A process in which the present state of the system depends on the past n states may be described in terms of n first-order Markov processes and is known as an n th-order Markov process. The term ‘Markov process’ usually refers to a first-order process.

Example 3.21

A simple example of a Markov process is a first-order autoregressive process (Figure 3.21) defined as

$$x(m) = ax(m-1) + e(m) \quad (3.146)$$

In Equation (3.146), $x(m)$ depends on the previous value $x(m-1)$ and the input $e(m)$. The conditional pdf of $x(m)$ given the previous sample value can be expressed as

$$\begin{aligned} f_X[x(m) | x(m-1), \dots, x(m-N)] &= f_X[x(m) | x(m-1)] \\ &= f_E[e(m) = x(m) - ax(m-1)] \end{aligned} \quad (3.147)$$

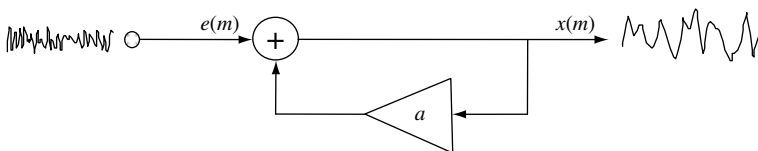


Figure 3.21 A first-order autoregressive (Markov) process.

where $f_E[e(m)]$ is the pdf of the input signal. Assuming that input $e(m)$ is a zero-mean Gaussian process with variance σ_e^2 , we have

$$\begin{aligned} f_X[x(m)|x(m-1), \dots, x(m-N)] &= f_X[x(m)|x(m-1)] \\ &= f_E[x(m) - ax(m-1)] \\ &= \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left\{-\frac{1}{2\sigma_e^2} [x(m) - ax(m-1)]^2\right\} \end{aligned} \quad (3.148)$$

When the input to a Markov model is a Gaussian process the output is known as a Gauss-Markov process.

3.7.9 MARKOV CHAIN PROCESSES

A discrete-time Markov process, $x(m)$, with N allowable states may be modelled by a Markov chain of N states (Figure 3.22). Each state can be associated with one of the N values that $x(m)$ may assume. In a Markov chain, the Markovian property is modelled by a set of state transition probabilities defined as

$$a_{ij}(m-1, m) = \text{Prob}[x(m) = j | x(m-1) = i] \quad (3.149)$$

where $a_{ij}(m-1, m)$ is the probability that at time $m-1$ the process is in the state i and then at time m it moves to state j . In Equation (3.144), the transition probability is expressed in a general time-dependent form. The marginal probability that a Markov process is in the state j at time m , $P_j(m)$, can be expressed as

$$P_j(m) = \sum_{i=1}^N P_i(m-1) a_{ij}(m-1, m) \quad (3.150)$$

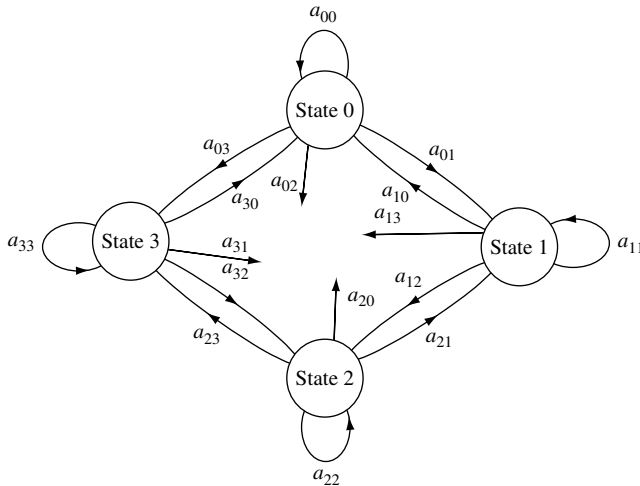


Figure 3.22 A Markov chain model of a four-state discrete-time Markov process.

A Markov chain is defined by the following set of parameters:

- number of states N ;
- state probability vector

$$\mathbf{p}^T(m) = [p_1(m), p_2(m), \dots, p_N(m)]$$

- the state transition matrix

$$A(m-1, m) = \begin{bmatrix} a_{11}(m-1, m) & a_{12}(m-1, m) & \dots & a_{1N}(m-1, m) \\ a_{21}(m-1, m) & a_{22}(m-1, m) & \dots & a_{2N}(m-1, m) \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1}(m-1, m) & a_{N2}(m-1, m) & \dots & a_{NN}(m-1, m) \end{bmatrix}$$

3.7.9.1 Homogenous and Inhomogeneous Markov Chains

A Markov chain with time-invariant state transition probabilities is known as a homogenous Markov chain. For a homogenous Markov process, the probability of a transition from state i to state j of the process is independent of the time of the transition m , as expressed in the following equation:

$$\text{Prob}[x(m) = j | x(m-1) = i] = a_{ij}(m-1, m) = a_{ij} \quad (3.151)$$

Homogeneous Markov chains have time-dependent transition probabilities. In most applications of Markov chains, homogenous models are used because they usually provide an adequate model of the signal process, and because homogenous Markov models are easier to train and use. Markov models are considered in Chapter 5.

3.7.10 GAMMA PROBABILITY DISTRIBUTION

The gamma pdf is defined as

$$\text{gamma}(x, a, b) = \begin{cases} \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-x/b} & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.152)$$

where a and b are both greater than zero and $\Gamma(a)$ is defined as

$$\Gamma(a) = \int_0^{\infty} x^{a-1} e^{-x} dx \quad (3.153)$$

Gamma pdf is sometimes used in modelling speech and image signals.

3.7.11 RAYLEIGH PROBABILITY DISTRIBUTION

The Rayleigh pdf is defined as

$$p(x) = \begin{cases} \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (3.154)$$

Rayleigh pdf is often employed to describe the amplitude spectrum of signals. In mobile communication the channel fading is usually modelled with a Rayleigh distribution.

3.7.12 LAPLACIAN PROBABILITY DISTRIBUTION

A Laplacian pdf is defined as

$$p(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right) \quad (3.155)$$

where σ is the standard deviation. Speech signal samples in time domain have a distribution that can be approximated by a Laplacian pdf. Laplacian pdf is also used for modelling image signals.

3.8 TRANSFORMATION OF A RANDOM PROCESS

In this section we consider the effect of filtering or transformation of a random process on its probability density function. Figure 3.23 shows a generalised mapping operator $h(\cdot)$ that transforms a random input process X into an output process Y . The input and output signals $x(m)$ and $y(m)$ are realisations of the random processes X and Y , respectively. If $x(m)$ and $y(m)$ are both discrete-valued such that $x(m) \in \{x_1, \dots, x_N\}$ and $y(m) \in \{y_1, \dots, y_M\}$ then we have

$$P_Y [y(m) = y_j] = \sum_{x_i \rightarrow y_j} P_X [x(m) = x_i] \quad (3.156)$$

where the summation is taken over all values of $x(m)$ that map to $y(m) = y_j$. Now consider the transformation of a discrete-time, continuous-valued, process. The probability that the output process Y has a value in the range $y(m) < Y < y(m) + \Delta y$ is

$$\text{Prob}[y(m) < Y < y(m) + \Delta y] = \int_{x(m) | y(m) < Y < y(m) + \Delta y} f_X [x(m)] dx(m) \quad (3.157)$$

where the integration is taken over all the values of $x(m)$ that yield an output in the range $y(m)$ to $y(m) + \Delta y$.

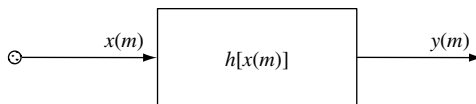


Figure 3.23 Transformation of a random process $x(m)$ to an output process $y(m)$.

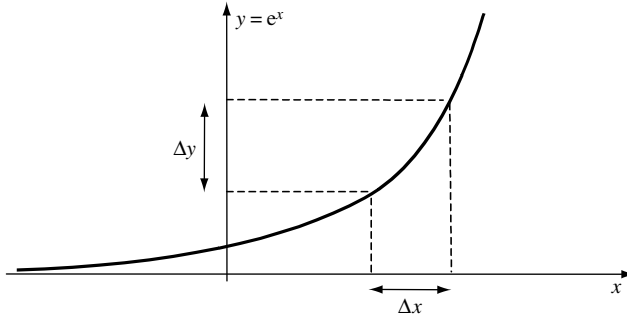


Figure 3.24 An example of a monotonic one-to-one mapping.

3.8.1 MONOTONIC TRANSFORMATION OF RANDOM PROCESSES

Now for a monotonic one-to-one transformation, $y(m) = h[x(m)]$ (as in Figure 3.24), Equation (3.157) becomes

$$\text{Prob}[y(m) < Y < y(m) + \Delta y] = \text{Prob}[x(m) < X < x(m) + \Delta x] \quad (3.158)$$

or, in terms of the cumulative distribution functions,

$$F_Y[y(m) + \Delta y] - F_Y[y(m)] = F_X[x(m) + \Delta x] - F_X[x(m)] \quad (3.159)$$

Multiplication of the left-hand side of Equation (3.159) by $\Delta y/\Delta y$ and the right-hand side by $\Delta x/\Delta x$ and re-arrangement of the terms yields

$$\frac{F_Y[y(m) + \Delta y] - F_Y[y(m)]}{\Delta y} = \frac{\Delta x}{\Delta y} \frac{F_X[x(m) + \Delta x] - F_X[x(m)]}{\Delta x} \quad (3.160)$$

Now as the intervals Δx and Δy tend to zero, Equation (3.160) becomes

$$f_Y[y(m)] = \left| \frac{\partial x(m)}{\partial y(m)} \right| f_X[x(m)] \quad (3.161)$$

where $f_Y[y(m)]$ is the probability density function. In Equation (3.161), substitution of $x(m) = h^{-1}[y(m)]$ yields

$$f_Y[y(m)] = \left| \frac{\partial h^{-1}[y(m)]}{\partial y(m)} \right| f_X\{h^{-1}[y(m)]\} \quad (3.162)$$

Equation (3.163) gives the pdf of the output signal in terms of the pdf of the input signal and the transformation.

Example 3.22: Probability Density of Frequency Spectrum – Cartesian to Polar Transformation

Consider the k th spectral component of the discrete Fourier transform of a discrete-time Gaussian distributed variable

$$X(k) = X_R(k) + jX_I(k) = r(k)e^{j\varphi(k)} \quad (3.163)$$

The Fourier transform is a linear operation, hence if the input is Gaussian it follows that the real part of the spectrum, $X_R(f)$, and the imaginary part of the spectrum, $X_I(f)$, are also Gaussian.

$$\begin{aligned} p(X_R, X_I) &= p(X_R) p(X_I) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{X_R^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{X_I^2}{2\sigma^2}\right) \\ &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{X_R^2 + X_I^2}{2\sigma^2}\right) \end{aligned} \quad (3.164)$$

Now the relation between transforming differential areas in Cartesian and polar co-ordinates is obtained as

$$dX_R dX_I = r dr d\varphi \quad (3.165)$$

Hence from Equation (3.161) we have

$$p(r, \varphi) = \frac{dX_R dX_I}{dr d\varphi} p(X_R, X_I) = r p(X_R, X_I) \quad (3.166)$$

From Equations (3.164) and (3.166) we have

$$p(r, \varphi) = \frac{r}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (3.167)$$

The probability of the phase φ can be obtained from

$$p(\varphi) = \int_0^\infty p(r, \varphi) dr = \frac{1}{2\pi} \int_0^\infty \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) dr = \frac{1}{2\pi} \quad (3.168)$$

Hence φ has a uniform distribution pdf with a probability of $1/2\pi$ and the pdf of r is Rayleigh distributed as

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & r \geq 0 \\ 0 & r < 0 \end{cases} \quad (3.169)$$

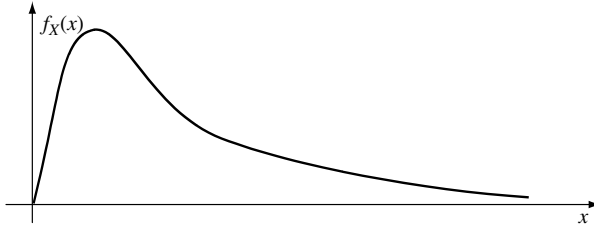


Figure 3.25 A log-normal probability density function.

Example 3.23

Transformation of a Gaussian process to a log-normal process. Log-normal pdfs are used for modelling positive-valued processes such as power spectra. If a random variable $x(m)$ has a Gaussian pdf, then the non-negative valued variable $y(m) = \exp[x(m)]$ has a log-normal distribution (Figure 3.25) obtained from Equation (3.161) as

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_x y(m)} \exp \left\{ -\frac{[\ln y(m) - \mu_x]^2}{2\sigma_x^2} \right\} \quad (3.170)$$

Conversely, if the input y to a logarithmic function has a log-normal distribution, then the output $x = \ln y$ is Gaussian. The mapping functions for translating the mean and variance of a log-normal distribution to a normal distribution can be derived as

$$\mu_x = \ln \mu_y - \frac{1}{2} \ln \left(1 + \frac{\sigma_y^2}{\mu_y^2} \right) \quad (3.171)$$

$$\sigma_x^2 = \ln \left(1 + \frac{\sigma_y^2}{\mu_y^2} \right) \quad (3.172)$$

(μ_x, σ_x^2) and (μ_y, σ_y^2) are the mean and variance of x and y , respectively. The inverse mapping relations for the translation of mean and variances of normal to log-normal variables are

$$\mu_y = \exp(\mu_x + \sigma_x^2/2) \quad (3.173)$$

$$\sigma_y^2 = \mu_x^2 [\exp(\sigma_x^2) - 1] \quad (3.174)$$

3.8.2 MANY-TO-ONE MAPPING OF RANDOM SIGNALS

Now consider the case when the transform $h(\cdot)$ is a nonmonotonic function such as that shown in Figure 3.26. Assuming that the equation $y(m) = h[x(m)]$ has K roots, there are K different values of $x(m)$ that map to the same $y(m)$. The probability that a realisation of the output process Y has a value in the range $y(m)$ to $y(m) + \Delta y$ is given by

$$\text{Prob}[y(m) < Y < y(m) + \Delta y] = \sum_{k=1}^K \text{Prob}[x_k(m) < X < x_k(m) + \Delta x_k] \quad (3.175)$$

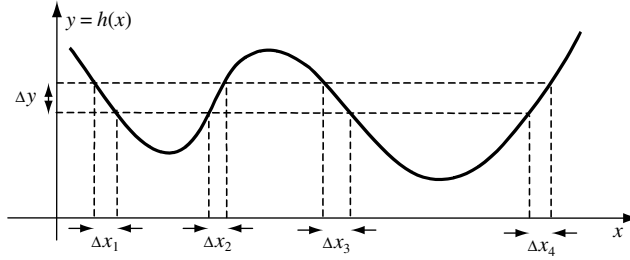


Figure 3.26 Illustration of a many-to-one transformation.

where x_k is the k th root of $y(m) = h[x(m)]$. Similar to the development in Section 3.7.1, Equation (3.175) can be written as

$$\frac{F_Y [y(m) + \Delta y] - F_Y [y(m)]}{\Delta y} \Delta y = \sum_{k=1}^K \frac{F_X [x_k(m) + \Delta x_k] - F_X [x_k(m)]}{\Delta x_k} \Delta x_k \tag{3.176}$$

Equation (3.176) can be rearranged as

$$\frac{F_Y [y(m) + \Delta y] - F_Y [y(m)]}{\Delta y} = \sum_{k=1}^K \frac{\Delta x_k}{\Delta y} \frac{F_X [x_k(m) + \Delta x_k] - F_X [x_k(m)]}{\Delta x_k} \tag{3.177}$$

Now as the intervals Δx and Δy tend to zero, Equation (3.177) becomes

$$\begin{aligned} f_Y [y(m)] &= \sum_{k=1}^K \left| \frac{\partial x_k(m)}{\partial y(m)} \right| f_X [x_k(m)] \\ &= \sum_{k=1}^K \frac{1}{|h'[x_k(m)]|} f_X [x_k(m)] \end{aligned} \tag{3.178}$$

where $h'[x_k(m)] = \partial h[x_k(m)] / \partial x_k(m)$. Note that, for a monotonic function, $K = 1$ and Equation (3.178) becomes the same as Equation (3.161). Equation (3.178) can be expressed as

$$f_Y [y(m)] = \sum_{k=1}^K |J[x_k(m)]|^{-1} f_X [x_k(m)] \tag{3.179}$$

where $J[x_k(m)] = h'[x_k(m)]$ is called the *Jacobian* of the transformation. For a multivariate transformation of a vector-valued process such as

$$\mathbf{y}(m) = \mathbf{H}[\mathbf{x}(m)] \tag{3.180}$$

the pdf of the output $\mathbf{y}(m)$ is given by

$$f_Y [\mathbf{y}(m)] = \sum_{k=1}^K |\mathbf{J}[\mathbf{x}_k(m)]|^{-1} f_X [\mathbf{x}_k(m)] \tag{3.181}$$

where $|\mathbf{J}(\mathbf{x})|$, the Jacobian of the transformation $\mathbf{H}(\cdot)$, is the determinant of a matrix of derivatives:

$$|\mathbf{J}(\mathbf{x})| = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_P} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_P}{\partial x_1} & \frac{\partial y_P}{\partial x_2} & \dots & \frac{\partial y_P}{\partial x_P} \end{vmatrix} \quad (3.182)$$

For a monotonic linear vector transformation such as

$$\mathbf{y} = \mathbf{H}\mathbf{x} \quad (3.183)$$

the pdf of \mathbf{y} becomes

$$f_Y(\mathbf{y}) = |\mathbf{J}|^{-1} f_X(\mathbf{H}^{-1}\mathbf{y}) \quad (3.184)$$

where $|\mathbf{J}|$ is the Jacobian of the transformation.

Example 3.24

The input–output relation of a $P \times P$ linear transformation matrix \mathbf{H} is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} \quad (3.185)$$

The Jacobian of the linear transformation \mathbf{H} is $|\mathbf{H}|$. Assume that the input \mathbf{x} is a zero-mean Gaussian P -variate process with a covariance matrix of Σ_{xx} and a probability density function given by:

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{P/2} |\Sigma_{xx}|^{1/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T \Sigma_{xx}^{-1} \mathbf{x}\right] \quad (3.186)$$

From Equations (3.168)–(3.169), the pdf of the output \mathbf{y} is given by

$$\begin{aligned} f_Y(\mathbf{y}) &= \frac{1}{(2\pi)^{P/2} |\Sigma_{xx}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^T \mathbf{H}^{-1T} \Sigma_{xx}^{-1} \mathbf{H}^{-1} \mathbf{y}\right) |\mathbf{H}|^{-1} \\ &= \frac{1}{(2\pi)^{P/2} |\Sigma_{xx}|^{1/2} |\mathbf{H}|} \exp\left(-\frac{1}{2}\mathbf{y}^T \Sigma_{yy}^{-1} \mathbf{y}\right) \end{aligned} \quad (3.187)$$

where $\Sigma_{yy} = \mathbf{H}\Sigma_{xx}\mathbf{H}^T$. Note that a linear transformation of a Gaussian process yields another Gaussian process.

3.8.2.1 Probabilistic Ranking: Web Page Citation Ranking and Indexing

Internet search engines sort and index the text information in many billions of web pages on the world wide web. A good set of search keywords will focus the search on the documents

and websites that contain the input keywords. However, the problem remains that the contents of many websites are not of the required quality. Furthermore, there are misleading websites containing popular keywords just to attract visitors and thus increase hit rates and advertising revenues.

For efficient information management, the websites and their information content need to be rank ordered using an objective quality measure. A well-established objective measure of quality of published information in any medium is citation ranking, which for a long time has been used as a ranking method in academic research. A map of hyperlinks and pointers to websites on the world wide web allows rapid calculation of a web page's rank in terms of citation. Page rank based on citation is a good way to organise the order of presentation of the results of an internet search.

3.8.2.2 Citation Ranking in Web Page Rank Calculation

Search engines usually find many web pages that contain the search keywords. The problem is how to present the web links containing the search keywords in a rank-ordered form, such that the rank of a page represents a measure of the quality of information on the page. The relevance of a web page, containing the search text string, can be determined from the following analysis of the web page:

- (1) the page title containing the search words is an indicator of the relevance of the topic of the page, but not of its quality;
- (2) the number of times the search words are mentioned in the web page is also an indicator;
- (3) the number of citation of the web page from other web pages is an objective indicator of the quality as perceived by web users;
- (4) each citation to a web page can be weighted by its importance, which itself is a weighted citation of the citation.

The simplest way to rank a web page is to count the total number of citation links pointing to that page and then divide this by the total number of citation links on the web. This method would rank a web page using a simple probability measure defined as the frequency of citation links. However, as with the tradition of academic research, a citation itself needs to be weighted by the quality of the citation source, i.e. by the citation ranking of the source itself. A weighted citation gives some approximation of a page's importance or quality, where each source of citation is weighted by its own citation ranking.

Let $PR(A)$ define the page rank for a web page A . Assume that page A has pages T_1, \dots, T_n pointing to it. The page rank of A can be defined as

$$PR(A) = (1 - d) + d [PR(T_1)/C(T_1) + \dots + PR(T_n)/C(T_n)]$$

where $C(T)$ is defined as the number of links going out of page T . The parameter d is a damping factor which can be set between 0 and 1; usually d is set to 0.85. Note that the page ranks form a probability distribution over web pages, so the sum of all web pages' page ranks will be 1. Page rank, $PR(A)$, can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.

3.9 SUMMARY

The theory of statistical processes is central to development of signal processing and communications systems. We began this chapter with some basic definitions of deterministic signals, random signals and random processes. A random process generates random signals, and the collection of all signals that can be generated by a random process is the space of the process. Probabilistic models and statistical measures, originally developed for random variables, were extended to model random signals. Although random signals are completely described in terms of probabilistic models, for many applications it may be sufficient to characterise a process in terms of a set of relatively simple statistics such as the mean, the autocorrelation function, the covariance and the power spectrum. Much of the theory and application of signal processing is concerned with the identification, extraction and utilisation of structures and patterns in a signal process. The correlation and its Fourier transform the power spectrum are particularly important because they can be used to identify the patterns in a stochastic process.

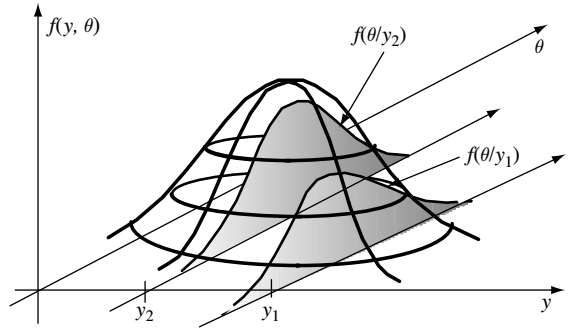
We considered the concepts of stationary, ergodic-stationary and nonstationary processes. The concept of a stationary process is central to the theory of linear time-invariant systems, and furthermore even nonstationary processes can be modelled by a chain of stationary sub-processes as described in Chapter 5 on hidden Markov models. For signal processing applications, a number of useful pdfs, including the Gaussian, the mixture Gaussian, the Markov and the Poisson process, were considered. These pdf models are extensively employed in the remainder of this book. Signal processing normally involves the filtering or transformation of an input signal to an output signal. We derived general expressions for the pdf of the output of a system in terms of the pdf of the input. We also considered some applications of stochastic processes for modelling random noise such as white noise, clutters, shot noise and impulsive noise.

BIBLIOGRAPHY

- Anderson O.D. (1976) *Time Series Analysis and Forecasting. The Box-Jenkins Approach*. Butterworth, London.
- Ayre A.J. (1972) *Probability and Evidence*. Columbia University Press, New York.
- Bartlett M.S. (1960) *Stochastic Processes*. Cambridge University Press, Cambridge.
- Box G.E.P and Jenkins G.M. (1976) *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA.
- Breiphol A.M. (1970) *Probabilistic System Analysis*. Wiley, New York.
- Carter G. (1987) Coherence and time delay estimation. *Proc. IEEE*, **75**(2): 236–255.
- Clark A.B. and Disney R.L. (1985) *Probability and Random Processes*, 2nd edn. Wiley, New York.
- Cooper G.R. and Mcgillum C.D. (1986) *Probabilistic Methods of Signal and System Analysis*. Holt, Rinehart and Winston, New York.
- Davenport W.B. and Root W.L. (1958) *Introduction to Random Signals and Noise*. McGraw-Hill, New York.
- Davenport W.B. and Wilbur B. (1970) *Probability and Random Processes: an Introduction for Applied Scientists and Engineers*. McGraw-Hill, New York.
- Einstein A. (1956) *Investigation on the Theory of Brownian Motion*. Dover, New York.
- Gardener W.A. (1986) *Introduction to Random Processes: With Application to Signals and Systems*. Macmillan, New York.
- Gauss C.F. (1963) *Theory of Motion of Heavenly Bodies*. Dover, New York.
- Helstrom C.W. (1991) *Probability and Stochastic Processes for Engineers*. Macmillan, New York.
- Isaacson D. and Masden R. (1976) *Markov Chains Theory and Applications*. Wiley, New York.
- Jeffrey H. (1961) *Scientific Inference*, 3rd edn. Cambridge University Press, Cambridge.

- Jeffrey H. (1973) *Theory of Probability*, 3rd edn. Clarendon Press, Oxford.
- Kay S.M. (1993) *Fundamentals of Statistical Signal Processing. Estimation Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Kendall M. and Stuart A. (1977) *The Advanced Theory of Statistics*. Macmillan, New York.
- Kolmogorov A.N. (1956) *Foundations of the Theory of Probability*. Chelsea Publishing Company, New York.
- Leon-Garcia A. (1994) *Probability and Random Processes for Electrical Engineering*. Addison Wesley, Reading, MA.
- Markov A.A. (1913) An example of statistical investigation in the text of *Eugen Onyegin* illustrating coupling of tests in chains. *Proc. Acad. Sci. St Petersburg VI Ser.*, **7**: 153–163.
- Meyer P.L. (1970) *Introductory Probability and Statistical Applications*. Addison-Wesley, Reading, MA.
- Papopulis A. (1977) *Signal Analysis*. McGraw-Hill, New York.
- Papopulis A. (1984) *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York.
- Parzen E. (1962) *Stochastic Processes*. Holden-Day, San Francisco, CA.
- Peebles P.Z. (1987) *Probability, Random Variables and Random Signal Principles*. McGraw-Hill, New York.
- Rao C.R. (1973) *Linear Statistical Inference and its Applications*. Wiley, New York.
- Rozanov Y.A. (1969) *Probability Theory: a Concise Course*. Dover, New York.
- Shanmugan K.S. and Breipohl A.M. (1988) *Random Signals: Detection, Estimation and Data Analysis*. Wiley, New York.
- Thomas J.B. (1988) *An Introduction to Applied Probability and Random Processes*. Huntington, Krieger Publishing, New York.
- Wozencraft J.M. and Jacobs I.M. (1965) *Principles of Communication Engineering*. Wiley, New York.

4



Bayesian Inference

4.1 Bayesian Estimation Theory: Basic Definitions	94
4.2 Bayesian Estimation	102
4.3 The Estimate–Maximise Method	116
4.4 Cramer–Rao Bound on the Minimum Estimator Variance	119
4.5 Design of Gaussian Mixture Models	121
4.6 Bayesian Classification	124
4.7 Modelling the Space of a Random Process	132
4.8 Summary	134
Bibliography	135

Bayesian estimation is the general framework for formulation of statistical inference problems. Inference is the act of deriving conclusions from evidence. In statistical estimation the value of a signal, or a parameter, is inferred from observations. In the prediction or estimation of a random process from a related signal, the Bayesian philosophy is based on combining the *evidence* contained in the signal with the *prior* knowledge of the probability distribution of the process.

Bayesian methodology includes the classical estimators such as maximum *a posteriori* (MAP), maximum-likelihood (ML), minimum mean square error (MMSE) and minimum mean absolute value of error (MAVE) as special cases. The hidden Markov model, widely used in pattern recognition, is an example of a Bayesian model. Bayesian inference is based on minimisation of the so-called Bayes' risk function, which includes a posterior model of the unknown parameters given the observation and a cost-of-error function.

This chapter begins with an introduction to the basic concepts of the estimation theory, and considers the statistical measures that are used to quantify the performance of an estimator.

We study Bayesian estimation methods and consider the effect of using a prior model on the mean and the variance of an estimate. The estimate–maximise (EM) method for the estimation of a set of unknown parameters from an incomplete observation is studied, and applied to the mixture Gaussian modelling of the space of a continuous random variable. This chapter concludes with Bayesian classification of discrete or finite-state signals, and the K-means clustering method.

4.1 BAYESIAN ESTIMATION THEORY: BASIC DEFINITIONS

Estimation theory is concerned with the determination of the best estimate of an unknown parameter vector from an observation signal, or the recovery of a clean signal degraded by noise and distortion. For example, given a noisy sine wave, we may be interested in estimating its basic parameters (i.e. amplitude, frequency and phase), or we may wish to recover the signal itself.

An estimator takes as input a set of noisy or incomplete observations, and, using a dynamic model (e.g. a linear predictive model) and/or a probabilistic model (e.g. Gaussian model) of the process, estimates the unknown parameters. The estimation accuracy depends on the available information and on the efficiency of the estimator. In this chapter, the Bayesian estimation of stationary parameters is studied. The modelling and estimation of nonstationary finite-state processes is covered in the next chapter.

Bayesian theory is a general inference framework. In the estimation or prediction of the state of a process, the Bayesian method employs both the evidence contained in the observation signal and prior knowledge of the probability distribution of the process. Figure 4.1 illustrates that the Bayesian method is the general framework for the derivation of the main statistical estimation methods.

Consider the estimation of the value of a random parameter vector θ , given a related observation vector y . From Bayes’ rule the posterior pdf of the parameter vector θ given

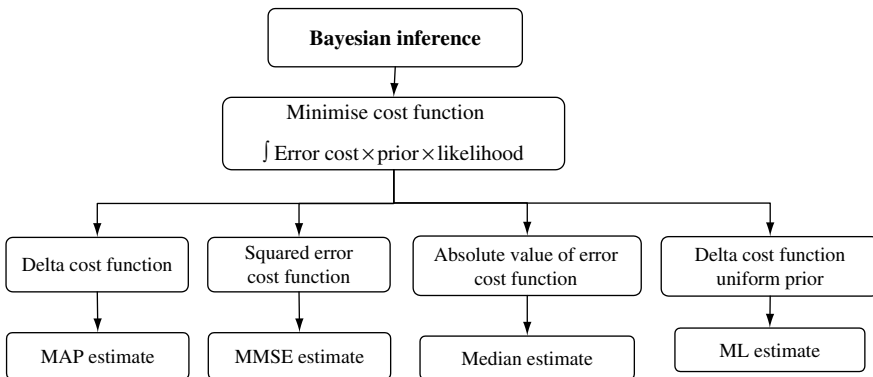


Figure 4.1 Bayesian inference involves a cost function, a prior function and a likelihood function. Other estimation methods can be considered as special cases of Bayesian estimation.

\mathbf{y} , $f_{\Theta|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y})$, can be expressed as

$$f_{\Theta|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y}) = \frac{f_{\mathbf{Y}|\Theta}(\mathbf{y}|\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})}{f_{\mathbf{Y}}(\mathbf{y})} \quad (4.1)$$

where for a given observation, $f_{\mathbf{Y}}(\mathbf{y})$ is a constant and has only a normalising effect. Thus there are two variable terms in Equation (4.1): one term $f_{\mathbf{Y}|\Theta}(\mathbf{y}|\boldsymbol{\theta})$ is the *likelihood* that the observation signal \mathbf{y} was generated by the parameter vector $\boldsymbol{\theta}$ and the second term is the *prior* probability of the parameter vector having a value of $\boldsymbol{\theta}$. Hence

$$\text{Posterior probability} \propto \text{likelihood} \times \text{prior}$$

The relative influence of the likelihood pdf $f_{\mathbf{Y}|\Theta}(\mathbf{y}|\boldsymbol{\theta})$ and the prior pdf $f_{\Theta}(\boldsymbol{\theta})$ on the posterior pdf $f_{\Theta|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y})$ depends on the shape of these functions, i.e. on how relatively peaked each pdf is. In general, the more peaked a pdf is, the more it will influence the outcome of the estimation process. Conversely, a uniform prior pdf will have no influence.

The remainder of this chapter is concerned with different forms of Bayesian estimation and its applications. First, in this section, some basic concepts of estimation theory are introduced.

4.1.1 DYNAMIC AND PROBABILITY MODELS IN ESTIMATION

Optimal estimation algorithms utilise *dynamic* and *probabilistic* models of the observation signals. A dynamic predictive model captures the correlation structure of a signal, and models the dependence of the present and future values of the signal on its past trajectory and the input stimulus. Examples of estimation methods employing dynamic models are linear prediction model and Kalman filters.

A statistical probability model characterises the *space* of random fluctuations of a signal in terms of its statistics, such as the mean and the covariance, and most completely in terms of a probability model. Conditional probability models, in addition to modelling the random fluctuations of a signal, can also model the dependence of the signal on its past values or on some other related process.

Dynamic and probability models can be combined; for example a finite-state model may be constructed from a combination of hidden Markov models (HMMs) and Kalman filters.

As an illustration, consider the estimation of a P -dimensional parameter vector $\boldsymbol{\theta} = [\theta_0, \theta_1, \dots, \theta_{P-1}]$ from a noisy observation vector $\mathbf{y} = [y(0), y(1), \dots, y(N-1)]$ modelled as

$$\mathbf{y} = \mathbf{x} + \mathbf{n} = h(\boldsymbol{\theta}, \mathbf{e}) + \mathbf{n} \quad (4.2)$$

where, as illustrated in Figure 4.2, it is assumed that the clean signal, \mathbf{x} , is the output of a predictive model, $h(\cdot)$, with a random input \mathbf{e} and parameter vector $\boldsymbol{\theta}$, and \mathbf{n} is an additive random noise process. In Figure 4.2, the distributions of the random noise, \mathbf{n} , the random input, \mathbf{e} , and the parameter vector, $\boldsymbol{\theta}$, are modelled by probability density functions, $f_{\mathbf{N}}(\mathbf{n})$, $f_{\mathbf{E}}(\mathbf{e})$ and $f_{\Theta}(\boldsymbol{\theta})$, respectively. The pdf model most often used is the Gaussian model.

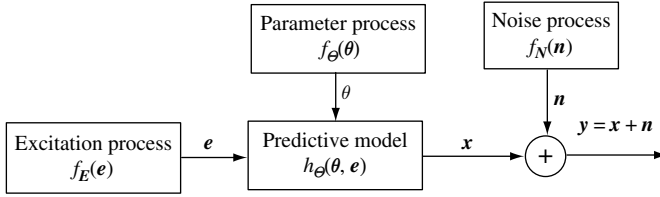


Figure 4.2 A random process y is described in terms of a predictive model $h(\cdot)$, and statistical models $f_E(\cdot)$, $f_\theta(\cdot)$ and $f_N(\cdot)$.

Predictive and statistical models of a process *guide* the estimator towards the set of values of the unknown parameters that is most consistent with both the prior distribution of the model parameters and the observation. In general, the more modelling information is used in an estimation process, the better the results, provided that the models are an accurate characterisation of the observation and the parameter process. The drawback is that, if the models are not accurate, more harm than good may result.

4.1.2 PARAMETER SPACE AND SIGNAL SPACE

Consider a random process with a parameter vector θ . For example, each instance of θ could be the parameter vector for a dynamic model of a speech sound or a musical note. The parameter space of a process θ is the collection of all the values that the parameter vector θ can assume.

The parameters of a random process determine the ‘characteristics’ (i.e. the mean, the variance, the power spectrum, etc.) of the signals generated by the process. As the process parameters change, so do the characteristics of the signals generated by the process. Each value of the parameter vector θ of a process has an associated signal space \mathbf{Y} ; this is the collection of all the signal realisations of the process with the parameter value θ .

For example, consider a three-dimensional vector-valued Gaussian process with parameter vector $\theta = [\mu, \Sigma]$, where μ is the mean vector and Σ is the covariance matrix of the Gaussian process. Figure 4.3 illustrates three mean vectors in a three-dimensional parameter space.

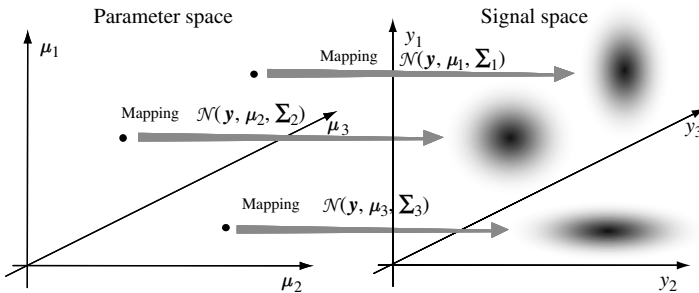


Figure 4.3 Illustration of three points in the parameter space of a Gaussian process and the associated signal spaces; for simplicity the variances are not shown in parameter space.

Also shown is the signal space associated with each parameter. As shown, the signal space of each parameter vector of a Gaussian process contains an infinite number of points, centred on the mean vector $\boldsymbol{\mu}$, and with a spatial volume and orientation that is determined by the covariance matrix $\boldsymbol{\Sigma}$. For simplicity, the variances are not shown in the parameter space, although they are evident in the shape of the Gaussian signal clusters in the signal space.

4.1.3 PARAMETER ESTIMATION AND SIGNAL RESTORATION

Parameter estimation and signal restoration are closely related problems. The main difference is due to the rapid fluctuations of most signals in comparison with the relatively slow variations of most parameters. For example, speech sounds fluctuate at speeds of up to 20 kHz, whereas the underlying vocal tract and pitch parameters vary at a relatively lower rate of less than 100 Hz. This observation implies that normally more averaging can be done in parameter estimation than in signal restoration.

As a simple example, consider a signal observed in a zero-mean random noise process. Assume we wish to estimate (a) the average of the clean signal and (b) the clean signal itself. As the observation length increases, the estimate of the signal mean approaches the mean value of the clean signal, whereas the estimate of the clean signal samples depends on the correlation structure of the signal and the signal-to-noise ratio as well as on the estimation method used.

As a further example, consider the interpolation of a sequence of lost samples of a signal given N recorded samples, as illustrated in Figure 4.4. Assume that an autoregressive (AR) process is used to model the signal as

$$y = X\boldsymbol{\theta} + e + n \tag{4.3}$$

where y is the observation signal, X is the signal matrix, $\boldsymbol{\theta}$ is the AR parameter vector, e is the random input of the AR model and n is the random noise. Using Equation (4.3), the signal restoration process involves the estimation of both the model parameter vector $\boldsymbol{\theta}$ and the random input e for the lost samples. Assuming the parameter vector $\boldsymbol{\theta}$ is time-invariant, the estimate of $\boldsymbol{\theta}$ can be averaged over the entire N observation samples and, as N becomes infinitely large, a consistent estimate should approach the true parameter value. The difficulty in signal interpolation is that the underlying excitation e of the signal x is purely random and, unlike $\boldsymbol{\theta}$, it cannot be estimated through an averaging operation.

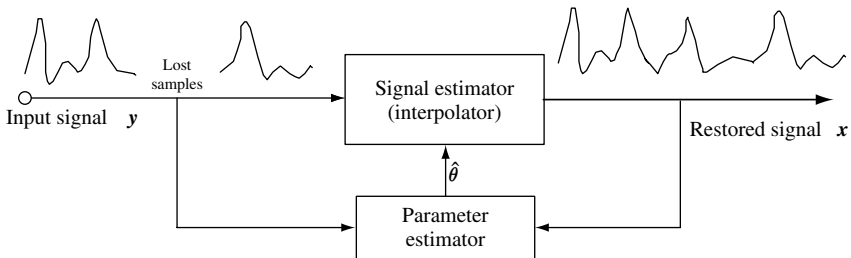


Figure 4.4 Illustration of signal restoration using a parametric model of the signal.

4.1.4 PERFORMANCE MEASURES AND DESIRABLE PROPERTIES OF ESTIMATORS

In estimation of a parameter vector θ from N observation samples \mathbf{y} , a set of performance measures is used to quantify and compare the characteristics of different estimators. In general, an estimate of a parameter vector is a function of the observation vector \mathbf{y} , the length of the observation N and the process model \mathcal{M} . This dependence may be expressed as

$$\hat{\theta} = f(\mathbf{y}, N, \mathcal{M}) \quad (4.4)$$

Different parameter estimators produce different results depending on the estimation method, utilisation of the observation and the influence of the prior information. Owing to the randomness of the observations, even the same estimator would produce different results with different observations from the same process. Therefore an estimate is itself a random variable, it has a mean and a variance, and it may be described by a pdf. However, for most cases, it is sufficient to characterise an estimator in terms of the mean and the variance of the estimation error. The most commonly used performance measures for an estimator are the following:

- (1) *Expected value* of estimate – $\mathcal{E}[\hat{\theta}]$;
- (2) *Bias* of estimate – $\mathcal{E}[\hat{\theta} - \theta] = \mathcal{E}[\hat{\theta}] - \theta$;
- (3) *Covariance* of estimate – $\text{Cov}[\hat{\theta}] = \mathcal{E}[(\hat{\theta} - \mathcal{E}[\hat{\theta}])(\hat{\theta} - \mathcal{E}[\hat{\theta}])^T]$.

Optimal estimators aim for zero bias and minimum estimation error covariance. The desirable properties of an estimator can be listed as follows:

- (1) *Unbiased estimator* – an estimator of θ is unbiased if the expectation of the estimate is equal to the true parameter value,

$$\mathcal{E}[\hat{\theta}] = \theta \quad (4.5)$$

An estimator is *asymptotically unbiased* if, for increasing length of observations N , we have

$$\lim_{N \rightarrow \infty} \mathcal{E}[\hat{\theta}] = \theta \quad (4.6)$$

- (2) *Efficient estimator* – an unbiased estimator of θ is an efficient estimator if it has the smallest covariance matrix compared with all other unbiased estimates of θ ,

$$\text{Cov}[\hat{\theta}_{\text{Efficient}}] \leq \text{Cov}[\hat{\theta}] \quad (4.7)$$

where $\hat{\theta}$ is any other estimate of θ .

- (3) *Consistent estimator* – an estimator is consistent if the estimate improves with the increasing length of the observation N , such that the estimate $\hat{\theta}$ converges probabilistically to the true value θ as N becomes infinitely large:

$$\lim_{N \rightarrow \infty} P[|\hat{\theta} - \theta| > \varepsilon] = 0 \quad (4.8)$$

where ε is arbitrary small.

Example 4.1

Consider the bias in the time-averaged estimates of the mean μ_y and the variance σ_y^2 of N observation samples $[y(0), \dots, y(N-1)]$, of an ergodic random process, given as

$$\hat{\mu}_y = \frac{1}{N} \sum_{m=0}^{N-1} y(m) \tag{4.9}$$

$$\hat{\sigma}_y^2 = \frac{1}{N} \sum_{m=0}^{N-1} [y(m) - \hat{\mu}_y]^2 \tag{4.10}$$

It is easy to show that $\hat{\mu}_y$ is an unbiased estimate, since

$$\mathcal{E} [\hat{\mu}_y] = \frac{1}{N} \sum_{m=0}^{N-1} \mathcal{E} [y(m)] = \mu_y \tag{4.11}$$

The expectation of the estimate of the variance can be expressed as

$$\begin{aligned} \mathcal{E} [\hat{\sigma}_y^2] &= \mathcal{E} \left\{ \frac{1}{N} \sum_{m=0}^{N-1} \left[y(m) - \frac{1}{N} \sum_{k=0}^{N-1} y(k) \right]^2 \right\} \\ &= \sigma_y^2 - \frac{2}{N} \sigma_y^2 + \frac{1}{N} \sigma_y^2 \\ &= \sigma_y^2 - \frac{1}{N} \sigma_y^2 \end{aligned} \tag{4.12}$$

From Equation (4.12), the bias in the estimate of the variance is inversely proportional to the signal length N , and vanishes as N tends to infinity; hence the estimate is asymptotically unbiased. In general, the bias and the variance of an estimate decrease with increasing number of observation samples, N , and with improved modelling. Figure 4.5 illustrates the general dependence of the distribution and the bias and the variance of an asymptotically unbiased estimator on the number of observation samples, N .

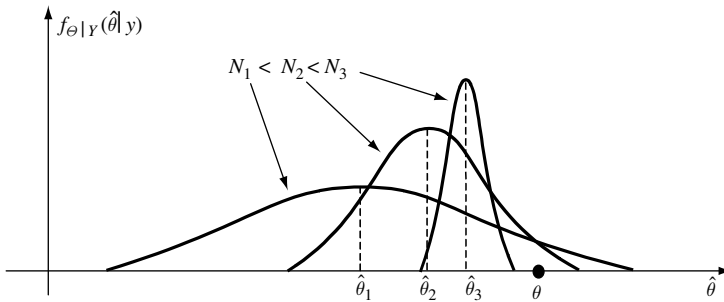


Figure 4.5 Illustration of the decrease in the bias and variance of an asymptotically unbiased estimate of the parameter θ with increasing length of observation.

4.1.5 PRIOR AND POSTERIOR SPACES AND DISTRIBUTIONS

The Bayesian inference method weights the likelihood that a value of parameter θ underlies an observation \mathbf{y} , $f(\mathbf{y}|\theta)$, with the prior probability of the value of the parameters $f(\theta)$. We may say that there is some similarity between Bayesian inference and human cognitive inference in that we often weight evidence with prior experience or disposition.

The *prior space* of a signal or a parameter vector is the collection of all possible values that the signal or the parameter vector can assume. Within the prior space, all the values of a parameter vector may have the same probability, in which case the prior space would have a uniform probability distribution, or the probability of the parameter vectors may have a nonuniform distribution. If the prior distribution is nonuniform (e.g. Gaussian, gamma, etc.), then the nonuniform prior can be used to weight the inference drawn from the observation; this would give more weight to the values that have a higher prior probability of occurrence.

The *evidence* of the value of a signal \mathbf{x} , or a parameter vector θ , is contained in the observation signal \mathbf{y} , which is used in a likelihood or cost function from which the signal or the parameter vector is estimated. For example a noisy speech signal, \mathbf{y} , may be used to obtain an estimate of the clean speech \mathbf{x} and/or the parameter vector θ of a linear prediction of model of speech.

The *posterior signal or parameter space* is the subspace of all the likely values of a signal \mathbf{x} , or a parameter vector θ , that are consistent with *both* the *prior* information on signal \mathbf{x} (or parameter θ) and the *evidence* contained in the observation \mathbf{y} . The significance of posterior probability is that the likelihood, $f_{\mathbf{Y}|\theta}(\mathbf{y}|\theta)$, of each value of a parameter θ is weighted with the prior probability of that value, $f_{\theta}(\theta)$.

For example, the likelihood that a variable such as tomorrow's weather, θ , takes a particular value/state given some meteorological observations \mathbf{y} , can be weighted with the prior likelihood of the weather states (irrespective of the observation), which itself could be obtained from previous years' weather data and would be also conditional on the time of the year.

Consider a random process with a parameter space Θ , observation space \mathbf{Y} and a joint pdf, $f_{\mathbf{Y},\theta}(\mathbf{y}, \theta)$. From the Bayes' rule the posterior pdf of the parameter vector, θ , given an observation vector \mathbf{y} , $f_{\theta|\mathbf{Y}}(\theta|\mathbf{y})$, can be expressed as

$$f_{\theta|\mathbf{Y}}(\theta|\mathbf{y}) = \frac{f_{\mathbf{Y}|\theta}(\mathbf{y}|\theta)f_{\theta}(\theta)}{f_{\mathbf{Y}}(\mathbf{y})} \quad (4.13)$$

where, for a given observation vector \mathbf{y} , the pdf $f_{\mathbf{Y}}(\mathbf{y})$ is a constant and has only a normalizing effect. From Equation (4.13), the posterior pdf is proportional to the weighted likelihood; that is the product of the likelihood $f_{\mathbf{Y}|\theta}(\mathbf{y}|\theta)$ that the observation \mathbf{y} was generated by the parameter vector θ , and the prior pdf $f_{\theta}(\theta)$. The prior pdf gives the unconditional parameter distribution *averaged* over the entire observation space as

$$f_{\theta}(\theta) = \int_{\mathbf{Y}} f_{\mathbf{Y},\theta}(\mathbf{y}, \theta) d\mathbf{y} \quad (4.14)$$

For most applications, it is relatively convenient to obtain the likelihood function, $f_{\mathbf{Y}|\theta}(\mathbf{y}|\theta)$. The *prior* pdf *influences* the inference drawn from the likelihood function by weighting it

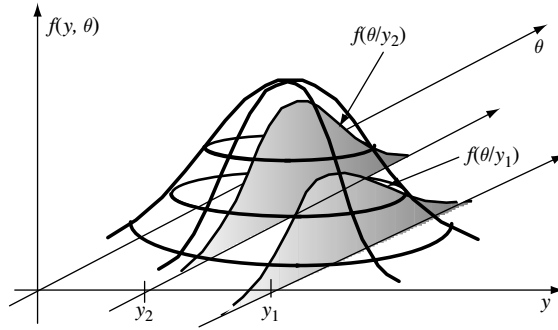


Figure 4.6 Illustration of joint distribution of signal y and parameter θ and the posterior distribution of θ given y .

with $f_{\theta}(\theta)$. The influence of the prior is particularly important for short-length and/or noisy observations, where the confidence in the estimate is limited by the lack of a sufficiently long observation and by the noise. The influence of the prior on the bias and the variance of an estimate are considered in Section 4.4.1.

A prior knowledge of the signal distribution can be used to confine the estimate to the prior signal space. The observation then guides the estimator to focus on the posterior space, that is, the subspace consistent with both the prior belief and the evidence contained in the observation. Figure 4.6 illustrates the joint pdf of a scalar signal $y(m)$ and a scalar parameter θ . As shown, an observation y cuts a posterior pdf $f_{\theta|Y}(\theta|y)$ through the joint distribution.

Example 4.2

A noisy signal vector of length N samples is modelled as

$$\mathbf{y}(m) = \mathbf{x}(m) + \mathbf{n}(m) \tag{4.15}$$

Assume that the signal $\mathbf{x}(m)$ is Gaussian with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the noise $\mathbf{n}(m)$ is also Gaussian with mean vector $\boldsymbol{\mu}_n$ and covariance matrix $\boldsymbol{\Sigma}_{nn}$. The signal and noise pdfs model the prior spaces of the signal and the noise, respectively. Given an observation vector $\mathbf{y}(m)$, the underlying signal $\mathbf{x}(m)$ would have a likelihood distribution with a mean vector of $\mathbf{y}(m) - \boldsymbol{\mu}_n$ and a covariance matrix $\boldsymbol{\Sigma}_{nn}$, as shown in Figure 4.7. The likelihood function is given by

$$\begin{aligned} f_{Y|\mathbf{x}}[\mathbf{y}(m)|\mathbf{x}(m)] &= f_N[\mathbf{y}(m) - \mathbf{x}(m)] \\ &= \frac{1}{(2\pi)^{N/2} |\boldsymbol{\Sigma}_{nn}|^{1/2}} \\ &\quad \times \exp \left\{ -\frac{1}{2} ([\mathbf{y}(m) - \boldsymbol{\mu}_n] - \mathbf{x}(m))^T \boldsymbol{\Sigma}_{nn}^{-1} ([\mathbf{y}(m) - \boldsymbol{\mu}_n] - \mathbf{x}(m)) \right\} \end{aligned} \tag{4.16}$$

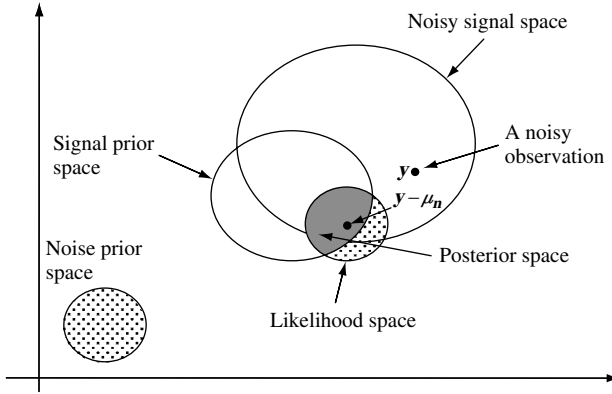


Figure 4.7 Sketch of a two-dimensional signal and noise spaces, and the likelihood and posterior spaces of a noisy observation, \mathbf{y} .

where the terms in the exponential function have been rearranged to emphasize the illustration of the likelihood space in Figure 4.7. Hence the posterior pdf can be expressed as

$$\begin{aligned}
 f_{X|Y}[\mathbf{x}(m)|\mathbf{y}(m)] &= \frac{f_{Y|X}[\mathbf{y}(m)|\mathbf{x}(m)]f_X[\mathbf{x}(m)]}{f_Y[\mathbf{y}(m)]} \\
 &= \frac{1}{f_Y[\mathbf{y}(m)]} \frac{1}{(2\pi)^N |\boldsymbol{\Sigma}_{nn}|^{1/2} |\boldsymbol{\Sigma}_{xx}|^{1/2}} \\
 &\quad \times \exp\left(-\frac{1}{2} \left\{ \underbrace{([\mathbf{y}(m) - \boldsymbol{\mu}_n] - \mathbf{x}(m))^T \boldsymbol{\Sigma}_{nn}^{-1} ([\mathbf{y}(m) - \boldsymbol{\mu}_n] - \mathbf{x}(m))}_{\text{Likelihood}} \right. \right. \\
 &\quad \left. \left. + \underbrace{[\mathbf{x}(m) - \boldsymbol{\mu}_x]^T \boldsymbol{\Sigma}_{xx}^{-1} [\mathbf{x}(m) - \boldsymbol{\mu}_x]}_{\text{Prior}} \right\} \right) \quad (4.17)
 \end{aligned}$$

For a two-dimensional signal and noise process, the prior spaces of the signal, the noise and the noisy signal are illustrated in Figure 4.7. The likelihood and the posterior spaces for a noisy observation vector, \mathbf{y} , are also illustrated in Figure 4.7. Note that the centre of the posterior space is obtained by subtracting the noise mean vector from the noisy signal vector. The clean signal is then somewhere within a subspace determined by the noise variance.

4.2 BAYESIAN ESTIMATION

The Bayesian estimation of a parameter vector, $\boldsymbol{\theta}$, is based on minimisation of a Bayesian risk function defined as an average cost-of-error function:

$$\begin{aligned}
\mathcal{R}(\hat{\boldsymbol{\theta}}) &= \mathcal{E}[C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})] \\
&= \int_{\boldsymbol{\theta}} \int_{\mathcal{Y}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\mathcal{Y}, \boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta}) \, d\mathbf{y} \, d\boldsymbol{\theta} \\
&= \int_{\boldsymbol{\theta}} \int_{\mathcal{Y}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\mathcal{Y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \, d\mathbf{y} \, d\boldsymbol{\theta}
\end{aligned} \tag{4.18}$$

where the cost-of-error function, $C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$, allows the appropriate weighting of the various outcomes to achieve desirable objective or subjective properties. The cost function can be chosen to associate a high cost with outcomes that are undesirable or disastrous. In Equation (4.19) the risk function is averaged over the space of all values of the parameter $\boldsymbol{\theta}$ and the observation \mathbf{y} .

For a given observation vector \mathbf{y} , $f_{\mathcal{Y}}(\mathbf{y})$ is a constant and has no effect on the risk-minimisation process. Hence Equation (4.18) may be written as a conditional risk function:

$$\mathcal{R}(\hat{\boldsymbol{\theta}}|\mathbf{y}) = \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\boldsymbol{\theta}|\mathcal{Y}}(\boldsymbol{\theta}|\mathbf{y}) \, d\boldsymbol{\theta} \tag{4.19}$$

The Bayesian estimate obtained as the minimum-risk parameter vector is given by

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg \min_{\hat{\boldsymbol{\theta}}} \mathcal{R}(\hat{\boldsymbol{\theta}}|\mathbf{y}) = \arg \min_{\hat{\boldsymbol{\theta}}} \left[\int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\boldsymbol{\theta}|\mathcal{Y}}(\boldsymbol{\theta}|\mathbf{y}) \, d\boldsymbol{\theta} \right] \tag{4.20}$$

Using Bayes's rule, Equation (4.20) can be written as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg \min_{\hat{\boldsymbol{\theta}}} \left[\int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\mathcal{Y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \right] \tag{4.21}$$

Assuming that the risk function is differentiable, and has a well-defined minimum, the Bayesian estimate can be obtained as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg \text{zero}_{\hat{\boldsymbol{\theta}}} \frac{\partial \mathcal{R}(\hat{\boldsymbol{\theta}}|\mathbf{y})}{\partial \hat{\boldsymbol{\theta}}} = \arg \text{zero}_{\hat{\boldsymbol{\theta}}} \left[\frac{\partial}{\partial \hat{\boldsymbol{\theta}}} \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\mathcal{Y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \right] \tag{4.22}$$

4.2.1 MAXIMUM A POSTERIORI ESTIMATION

The MAP estimate, $\hat{\boldsymbol{\theta}}_{\text{MAP}}$, is obtained as the parameter vector that maximises the posterior pdf, $f_{\boldsymbol{\theta}|\mathcal{Y}}(\boldsymbol{\theta}|\mathbf{y})$. The MAP estimate corresponds to a Bayesian estimate with a so-called uniform cost function (in fact, as shown in Figure 4.8, the cost function is notch-shaped), defined as

$$C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = 1 - \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \tag{4.23}$$

where $\delta(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ is the Kronecker delta function. Substitution of the cost function in the Bayesian risk equation yields

$$\begin{aligned}
\mathcal{R}_{\text{MAP}}(\hat{\boldsymbol{\theta}}|\mathbf{y}) &= \int_{\boldsymbol{\theta}} [1 - \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})] f_{\boldsymbol{\theta}|\mathcal{Y}}(\boldsymbol{\theta}|\mathbf{y}) \, d\boldsymbol{\theta} \\
&= 1 - f_{\boldsymbol{\theta}|\mathcal{Y}}(\hat{\boldsymbol{\theta}}|\mathbf{y})
\end{aligned} \tag{4.24}$$

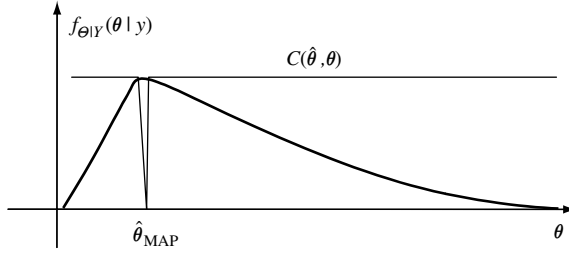


Figure 4.8 Illustration of the Bayesian cost function for the MAP estimate.

From Equation (4.24), the minimum Bayesian risk estimate corresponds to the parameter value where the posterior function attains a maximum. Hence the MAP estimate of the parameter vector, $\boldsymbol{\theta}$, is obtained from a minimisation of the risk Equation (4.24) or equivalently maximisation of the posterior function:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{MAP}} &= \arg \max_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\mathbf{y}) \\ &= \arg \max_{\boldsymbol{\theta}} [f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})f_{\boldsymbol{\theta}}(\boldsymbol{\theta})]\end{aligned}\quad (4.25)$$

4.2.2 MAXIMUM-LIKELIHOOD ESTIMATION

The ML estimate $\hat{\boldsymbol{\theta}}_{\text{ML}}$ is obtained as the parameter vector that maximises the likelihood function, $f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})$. The ML estimator corresponds to a Bayesian estimator with a notch-shaped cost function and a uniform-parameter prior pdf:

$$\begin{aligned}\mathcal{R}_{\text{ML}}(\hat{\boldsymbol{\theta}}|\mathbf{y}) &= \int_{\boldsymbol{\theta}} \underbrace{[1 - \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})]}_{\text{cost function}} \underbrace{f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}_{\text{likelihood}} \underbrace{f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}_{\text{prior}} d\boldsymbol{\theta} \\ &= \text{const.}[1 - f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\hat{\boldsymbol{\theta}})]\end{aligned}\quad (4.26)$$

where the prior function $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \text{const.}$

From a Bayesian perspective the main difference between the ML and MAP estimators is that the ML estimator assumes that the prior pdf of $\boldsymbol{\theta}$ is uniform. Note that a uniform prior, in addition to modelling genuinely uniform pdfs, is also used when the parameter prior pdf is unknown, or when the parameter is an unknown constant.

From Equation (4.26), it is evident that minimisation of the risk function is achieved by maximisation of the likelihood function:

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg \max_{\boldsymbol{\theta}} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \quad (4.27)$$

In practice it is convenient to maximise the log-likelihood function instead of the likelihood:

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \arg \max_{\boldsymbol{\theta}} \log f_{Y|\boldsymbol{\theta}}(\mathbf{Y}|\boldsymbol{\theta}) \quad (4.28)$$

The log-likelihood is usually chosen in practice because of the following properties:

- (1) the logarithm is a monotonic function, and hence the log-likelihood has the same turning points as the likelihood function;
- (2) the joint log-likelihood of a set of independent variables is the sum of the log-likelihood of individual variables; and
- (3) unlike the likelihood function, the log-likelihood has a dynamic range that does not cause computational under-flow.

Example 4.3: ML Estimation of the Mean and Variance of a Gaussian Process

Consider the problem of maximum likelihood estimation of the mean vector $\boldsymbol{\mu}_y$ and the covariance matrix $\boldsymbol{\Sigma}_{yy}$ of a P -dimensional Gaussian vector process from N observation vectors $[\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(N-1)]$. Assuming the observation vectors are uncorrelated, the pdf of the observation sequence is given by

$$f_Y[\mathbf{y}(0), \dots, \mathbf{y}(N-1)] = \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_{yy}|^{1/2}} \exp \left\{ -\frac{1}{2} [\mathbf{y}(m) - \boldsymbol{\mu}_y]^T \boldsymbol{\Sigma}_{yy}^{-1} [\mathbf{y}(m) - \boldsymbol{\mu}_y] \right\} \quad (4.29)$$

and the log-likelihood equation is given by

$$\ln f_Y[\mathbf{y}(0), \dots, \mathbf{y}(N-1)] = \sum_{m=0}^{N-1} \left\{ -\frac{P}{2} \ln(2\pi) - \frac{1}{2} \ln |\boldsymbol{\Sigma}_{yy}| - \frac{1}{2} [\mathbf{y}(m) - \boldsymbol{\mu}_y]^T \boldsymbol{\Sigma}_{yy}^{-1} [\mathbf{y}(m) - \boldsymbol{\mu}_y] \right\} \quad (4.30)$$

Taking the derivative of the log-likelihood equation with respect to the mean vector $\boldsymbol{\mu}_y$ yields

$$\frac{\partial \ln f_Y[\mathbf{y}(0), \dots, \mathbf{y}(N-1)]}{\partial \boldsymbol{\mu}_y} = \sum_{m=0}^{N-1} [2\boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\mu}_y - 2\boldsymbol{\Sigma}_{yy}^{-1} \mathbf{y}(m)] = 0 \quad (4.31)$$

From Equation (4.31), we have

$$\hat{\boldsymbol{\mu}}_y = \frac{1}{N} \sum_{m=0}^{N-1} \mathbf{y}(m) \quad (4.32)$$

To obtain the ML estimate of the covariance matrix we take the derivative of the log-likelihood equation with respect to $\boldsymbol{\Sigma}_{yy}^{-1}$:

$$\frac{\partial \ln f_Y[\mathbf{y}(0), \dots, \mathbf{y}(N-1)]}{\partial \boldsymbol{\Sigma}_{yy}^{-1}} = \sum_{m=0}^{N-1} \left\{ \frac{1}{2} \boldsymbol{\Sigma}_{yy} - \frac{1}{2} [\mathbf{y}(m) - \boldsymbol{\mu}_y][\mathbf{y}(m) - \boldsymbol{\mu}_y]^T \right\} = 0 \quad (4.33)$$

From Equation (4.31), we have an estimate of the covariance matrix as

$$\hat{\boldsymbol{\Sigma}}_{yy} = \frac{1}{N} \sum_{m=0}^{N-1} [\mathbf{y}(m) - \hat{\boldsymbol{\mu}}_y][\mathbf{y}(m) - \hat{\boldsymbol{\mu}}_y]^T \quad (4.34)$$

Example 4.4: ML and MAP Estimation of a Gaussian Random Parameters

Consider the estimation of a P -dimensional random parameter vector, $\boldsymbol{\theta}$, from an N -dimensional observation vector, \mathbf{y} . Assume that the relation between the signal vector \mathbf{y} and the parameter vector $\boldsymbol{\theta}$ is described by a linear model as

$$\mathbf{y} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e} \quad (4.35)$$

where \mathbf{e} is a random excitation input signal. The pdf of the parameter vector, $\boldsymbol{\theta}$, given an observation vector, \mathbf{y} , can be described, using Bayes's rule, as

$$f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{f_{\mathbf{y}}(\mathbf{y})} f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \quad (4.36)$$

Assuming that the matrix \mathbf{G} in Equation (4.35) is known, the likelihood of the signal \mathbf{y} given the parameter vector $\boldsymbol{\theta}$ is the pdf of the random vector \mathbf{e} :

$$f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) = f_{\mathbf{E}}(\mathbf{e} = \mathbf{y} - \mathbf{G}\boldsymbol{\theta}) \quad (4.37)$$

Now assume the input \mathbf{e} is a zero-mean, Gaussian-distributed, random process with a diagonal covariance matrix, and the parameter vector $\boldsymbol{\theta}$ is also a Gaussian process with mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}$. Therefore, the likelihood function, Equation (4.37), can be written as

$$f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) = f_{\mathbf{E}}(\mathbf{e}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left[-\frac{1}{2\sigma_e^2}(\mathbf{y} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{G}\boldsymbol{\theta})\right] \quad (4.38)$$

and

$$f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right] \quad (4.39)$$

The ML estimate obtained from maximisation of the log-likelihood function $\ln[f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is given by

$$\hat{\boldsymbol{\theta}}_{\text{ML}}(\mathbf{y}) = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{y} \quad (4.40)$$

To obtain the MAP estimate we first form the posterior distribution by substituting Equations (4.38) and (4.39) in Equation (4.36)

$$\begin{aligned} f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}) &= \frac{1}{f_{\mathbf{y}}(\mathbf{y})} \frac{1}{(2\pi\sigma_e^2)^{N/2}} \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}|^{1/2}} \\ &\times \exp\left[-\frac{1}{2\sigma_e^2}(\mathbf{y} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{G}\boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right] \end{aligned} \quad (4.41)$$

The MAP parameter estimate is obtained by differentiating the log-likelihood function, $\ln f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y})$, and setting the derivative to zero:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}}(\mathbf{y}) = (\mathbf{G}^T \mathbf{G} + \sigma_e^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1})^{-1} (\mathbf{G}^T \mathbf{y} + \sigma_e^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1} \boldsymbol{\mu}_{\boldsymbol{\theta}}) \quad (4.42)$$

Note that, as the covariance of the Gaussian-distributed parameter increases, or equivalently as $\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}}^{-1} \rightarrow 0$, the Gaussian prior tends to a uniform prior and the MAP solution

[Equation (4.42)] tends to the ML solution given by Equation (4.40). Conversely as the pdf of the parameter vector θ becomes peaked, i.e. as $\Sigma_{\theta\theta} \rightarrow 0$, the estimate tends towards μ_{θ} , the mean of the prior pdf.

4.2.3 MINIMUM MEAN SQUARE ERROR ESTIMATION

The Bayesian MMSE estimate is obtained as the parameter vector that minimises a mean square error cost function (Figure 4.9), defined as

$$\begin{aligned} \mathcal{R}_{\text{MMSE}}(\hat{\theta}|\mathbf{y}) &= \mathcal{E} \left[(\hat{\theta} - \theta)^2 | \mathbf{y} \right] \\ &= \int_{\theta} (\hat{\theta} - \theta)^2 f_{\theta|\mathbf{Y}}(\theta|\mathbf{y}) d\theta \end{aligned} \tag{4.43}$$

In the following, it is shown that *the Bayesian MMSE estimate is the conditional mean of the posterior pdf*. Assuming that the mean square error risk function is differentiable and has a well-defined minimum, the MMSE solution can be obtained by setting the gradient of the mean square error risk function to zero:

$$\frac{\partial \mathcal{R}_{\text{MMSE}}(\hat{\theta}|\mathbf{y})}{\partial \hat{\theta}} = 2\hat{\theta} \int_{\theta} f_{\theta|\mathbf{Y}}(\theta|\mathbf{y}) d\theta - 2 \int_{\theta} \theta f_{\theta|\mathbf{Y}}(\theta|\mathbf{y}) d\theta \tag{4.44}$$

Since the first integral on the right-hand side of Equation (4.44) is equal to 1, we have

$$\frac{\partial \mathcal{R}_{\text{MMSE}}(\hat{\theta}|\mathbf{y})}{\partial \hat{\theta}} = 2\hat{\theta} - 2 \int_{\theta} \theta f_{\theta|\mathbf{Y}}(d\theta|\mathbf{y}) d\theta \tag{4.45}$$

The MMSE solution is obtained by setting Equation (4.45) to zero:

$$\hat{\theta}_{\text{MMSE}}(\mathbf{y}) = \int_{\theta} \theta f_{\theta|\mathbf{Y}}(\theta|\mathbf{y}) d\theta \tag{4.46}$$

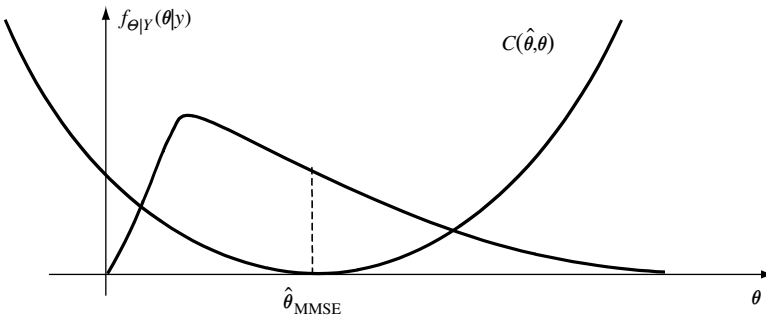


Figure 4.9 Illustration of the mean square error cost function and estimate.

For cases where we do not have pdf models of the parameter process $\boldsymbol{\theta}$ and the signal \mathbf{y} , the minimum mean square error (known as the least square error, LSE) estimate is obtained through minimisation of a mean square error function $\mathcal{E}[e^2(\boldsymbol{\theta}|\mathbf{y})]$:

$$\hat{\boldsymbol{\theta}}_{\text{LSE}} = \arg \min_{\boldsymbol{\theta}} \mathcal{E}[e^2(\boldsymbol{\theta}|\mathbf{y})] \quad (4.47)$$

where $e(\boldsymbol{\theta}|\mathbf{y})$ is an error signal. Least squared error estimation is considered in some depth in Chapter 6 on Wiener filters, in Chapter 7 on adaptive filters and in Chapter 8 on linear prediction. For a process with a Gaussian likelihood and a uniform parameter prior, the MMSE estimate is the same as the LSE estimate. The LSE estimation of Equation (4.47) does not use any prior knowledge of the distribution of the signals and the parameters. This can be considered as a strength of the LSE method in situations where the prior pdfs are unknown, but it can also be considered as a weakness in cases where fairly accurate models of the priors are available but not utilised. In the following example it is shown that the least squared error solution of Equation (4.47) is equivalent to ML solution when the signal has a Gaussian distribution.

Example 4.5

Consider the LSE estimation of the parameter vector $\boldsymbol{\theta}$ of a linear signal model from an observation signal vector \mathbf{y} given as

$$\mathbf{y} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e} \quad (4.48)$$

where \mathbf{G} is a matrix. The LSE estimate is obtained as the parameter vector at which the gradient of the mean squared error, $\mathbf{e}^T\mathbf{e}$, with respect to $\boldsymbol{\theta}$ is zero:

$$\left. \frac{\partial \mathbf{e}^T\mathbf{e}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} (\mathbf{y}^T\mathbf{y} - 2\boldsymbol{\theta}^T\mathbf{G}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{G}^T\mathbf{G}\boldsymbol{\theta}) \right|_{\boldsymbol{\theta}_{\text{LSE}}} = 0 \quad (4.49)$$

From Equation (4.49) the LSE parameter estimate is given by

$$\boldsymbol{\theta}_{\text{LSE}} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{y} \quad (4.50)$$

Note that for a Gaussian likelihood function, the LSE solution is the same as the ML solution of Equation (4.40).

4.2.4 MINIMUM MEAN ABSOLUTE VALUE OF ERROR ESTIMATION

The minimum MAVE estimate (Figure 4.10) is obtained through minimisation of a Bayesian risk function defined as

$$\mathcal{R}_{\text{MAVE}}(\hat{\boldsymbol{\theta}}|\mathbf{y}) = \mathcal{E} [|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|\mathbf{y}] = \int_{\boldsymbol{\theta}} |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}| f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \quad (4.51)$$

In the following it is shown that the minimum MAVE estimate is the median of the parameter process. Equation (4.51) can be expressed as

$$\mathcal{R}_{\text{MAVE}}(\hat{\boldsymbol{\theta}}|\mathbf{y}) = \int_{-\infty}^{\hat{\boldsymbol{\theta}}} [\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} + \int_{\hat{\boldsymbol{\theta}}}^{\infty} [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}] f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \quad (4.52)$$

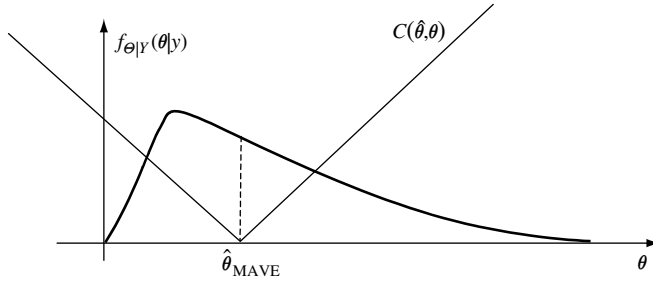


Figure 4.10 Illustration of mean absolute value of error cost function. Note that the MAVE estimate coincides with the conditional median of the posterior function.

Taking the derivative of the risk function with respect to $\hat{\theta}$ yields

$$\frac{\partial \mathcal{R}_{MAVE}(\hat{\theta}|\mathbf{y})}{\partial \hat{\theta}} = \int_{-\infty}^{\hat{\theta}} f_{\theta|Y}(\theta|\mathbf{y})d\theta - \int_{\hat{\theta}}^{\infty} f_{\theta|Y}(\theta|\mathbf{y})d\theta \tag{4.53}$$

The minimum absolute value of error is obtained by setting Equation (4.53) to zero:

$$\int_{-\infty}^{\hat{\theta}_{MAVE}} f_{\theta|Y}(\theta|\mathbf{y})d\theta = \int_{\hat{\theta}_{MAVE}}^{\infty} f_{\theta|Y}(\theta|\mathbf{y})d\theta \tag{4.54}$$

From Equation (4.54) we note the MAVE estimate is the median of the posterior density.

4.2.5 EQUIVALENCE OF THE MAP, ML, MMSE AND MAVE FOR GAUSSIAN PROCESSES WITH UNIFORM DISTRIBUTED PARAMETERS

Example 4.5 shows that, for a Gaussian-distributed process, the LSE estimate and the ML estimate are identical. Furthermore, Equation (4.42) for the MAP estimate of a Gaussian-distributed parameter shows that, as the parameter variance increases, or equivalently as the parameter prior pdf tends to a uniform distribution, the MAP estimate tends to the ML and LSE estimates. In general, for any symmetric distribution, centred round the maximum, the mode, the mean and the median are identical. Hence, for a process with a symmetric pdf, if the prior distribution of the parameter is uniform then the MAP, the ML, the MMSE and the MAVE parameter estimates are identical. Figure 4.11 illustrates a symmetric pdf, an asymmetric pdf and the relative positions of various estimates.

4.2.6 THE INFLUENCE OF THE PRIOR ON ESTIMATION BIAS AND VARIANCE

The use of a prior pdf introduces a bias in the estimate towards the range of parameter values with relatively high values of prior pdf, and reduces the variance of the estimate. To

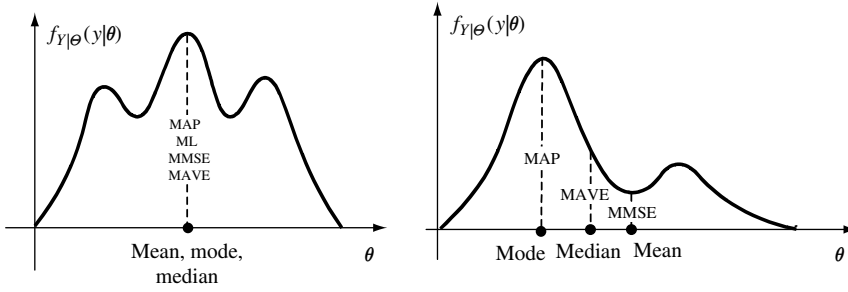


Figure 4.11 Illustration of a symmetric and an asymmetric pdf and their respective mode, mean and median and the relations to MAP, MAVE and MMSE estimates.

illustrate the effects of the prior pdf on the bias and the variance of an estimate, we consider the following examples in which the bias and the variance of the ML and the MAP estimates of the mean of a process are compared.

Example 4.6

Consider the ML estimation of a random scalar parameter θ , observed in a zero-mean additive white Gaussian noise (AWGN), $n(m)$, and expressed as

$$y(m) = \theta + n(m), \quad m = 0, \dots, N-1 \quad (4.55)$$

It is assumed that, for each realisation of the parameter θ , N observation samples are available. Note that, since the noise is assumed to be a zero-mean process, this problem is equivalent to estimation of the mean of the process $y(m)$. The likelihood of an observation vector $\mathbf{y} = [y(0), y(1), \dots, y(N-1)]$ and a parameter value of θ is given by

$$f_{Y|\theta}(\mathbf{y}|\theta) = \prod_{m=0}^{N-1} f_N[y(m) - \theta] = \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma_n^2} \sum_{m=0}^{N-1} [y(m) - \theta]^2 \right\} \quad (4.56)$$

From Equation (4.56) the log-likelihood function is given by

$$\ln f_{Y|\theta}(\mathbf{y}|\theta) = -\frac{N}{2} \ln(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2} \sum_{m=0}^{N-1} [y(m) - \theta]^2 \quad (4.57)$$

The ML estimate of θ , obtained by setting the derivative of $\ln f_{Y|\theta}(\mathbf{y}|\theta)$ to zero, is given by

$$\hat{\theta}_{\text{ML}} = \frac{1}{N} \sum_{m=0}^{N-1} y(m) = \bar{y} \quad (4.58)$$

where \bar{y} denotes the time average of $y(m)$. From Equation (4.56), we note that the ML solution is an unbiased estimate

$$\mathcal{E}[\hat{\theta}_{\text{ML}}] = \mathcal{E} \left[\frac{1}{N} \sum_{m=0}^{N-1} y(m) \right] = \mathcal{E} \left\{ \frac{1}{N} \sum_{m=0}^{N-1} [\theta + n(m)] \right\} = \theta \quad (4.59)$$

and the variance of the ML estimate is given by

$$\text{Var}[\hat{\theta}_{\text{ML}}] = \mathcal{E}[(\hat{\theta}_{\text{ML}} - \theta)^2] = \mathcal{E} \left\{ \left[\frac{1}{N} \sum_{m=0}^{N-1} y(m) - \theta \right]^2 \right\} = \frac{\sigma_n^2}{N} \quad (4.60)$$

Note that the variance of the ML estimate decreases with increasing length of observation.

Example 4.7: Estimation of a Uniformly Distributed Parameter Observed in AWGN

Consider the effects of using a uniform parameter prior on the mean and the variance of the estimate in Example 4.6. Assume that the prior for the parameter θ is given by

$$f_{\Theta}(\theta) = \begin{cases} 1/(\theta_{\text{max}} - \theta_{\text{min}}) & \theta_{\text{min}} \leq \theta \leq \theta_{\text{max}} \\ 0 & \text{otherwise} \end{cases} \quad (4.61)$$

as illustrated in Figure 4.12. From Bayes' rule, the posterior pdf of uniformly distributed is given by

$$\begin{aligned} f_{\Theta|Y}(\theta|y) &= \frac{1}{f_Y(y)} f_{Y|\Theta}(y|\theta) f_{\Theta}(\theta) \\ &= \begin{cases} \frac{1}{f_Y(y)} \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma_n^2} \sum_{m=0}^{N-1} [y(m) - \theta]^2 \right\}, & \theta_{\text{min}} \leq \theta \leq \theta_{\text{max}} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (4.62)$$

The MAP estimate is obtained by maximising the posterior pdf:

$$\hat{\theta}_{\text{MAP}}(y) = \begin{cases} \theta_{\text{min}} & \text{if } \hat{\theta}_{\text{ML}}(y) < \theta_{\text{min}} \\ \hat{\theta}_{\text{ML}}(y) & \text{if } \theta_{\text{min}} \geq \hat{\theta}_{\text{ML}}(y) \geq \theta_{\text{max}} \\ \theta_{\text{max}} & \text{if } \hat{\theta}_{\text{ML}}(y) > \theta_{\text{max}} \end{cases} \quad (4.63)$$

Note that the MAP estimate is constrained to the range $\theta_{\text{min}} - \theta_{\text{max}}$. This constraint is desirable and moderates the estimates that, due to say low signal-to-noise ratio, fall outside the range

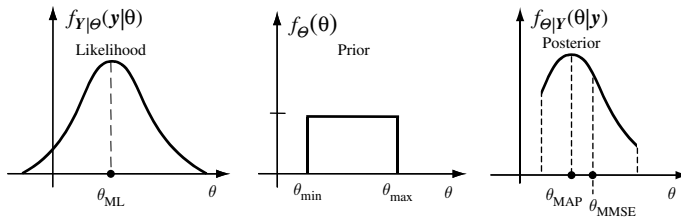


Figure 4.12 Illustration of the effects of a uniform prior on the estimate of a parameter observed in AWGN.

of possible values of θ . It is easy to see that the variance of an estimate constrained to a range of $\theta_{\min} - \theta_{\max}$ is less than the variance of the ML estimate in which there is no constraint on the range of the parameter estimate:

$$\text{Var}[\hat{\theta}_{\text{MAP}}] = \int_{\theta_{\min}}^{\theta_{\max}} (\hat{\theta}_{\text{MAP}} - \theta)^2 f_{Y|\theta}(y|\theta) dy \leq \text{Var}[\hat{\theta}_{\text{ML}}] = \int_{-\infty}^{\infty} (\hat{\theta}_{\text{ML}} - \theta)^2 f_{Y|\theta}(y|\theta) dy \quad (4.64)$$

Example 4.8: Estimation of a Gaussian-distributed Parameter Observed in AWGN

In this example, Figure 4.13, we consider the effect of a Gaussian prior on the mean and the variance of the MAP estimate. Assume that the parameter θ is Gaussian-distributed with a mean μ_θ and a variance σ_θ^2 as

$$f_\theta(\theta) = \frac{1}{(2\pi\sigma_\theta^2)^{1/2}} \exp\left[-\frac{(\theta - \mu_\theta)^2}{2\sigma_\theta^2}\right] \quad (4.65)$$

From Bayes rule the posterior pdf is given as the product of the likelihood and the prior pdfs as:

$$\begin{aligned} f_{\theta|Y}(\theta|y) &= \frac{1}{f_Y(y)} f_{Y|\theta}(y|\theta) f_\theta(\theta) \\ &= \frac{1}{f_Y(y)} \frac{1}{(2\pi\sigma_n^2)^{N/2} (2\pi\sigma_\theta^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma_n^2} \sum_{m=0}^{N-1} [y(m) - \theta]^2 - \frac{1}{2\sigma_\theta^2} (\theta - \mu_\theta)^2\right\} \end{aligned} \quad (4.66)$$

The maximum posterior solution is obtained by setting the derivative of the log-posterior function, $f_{\theta|Y}(\theta|y)$, with respect to θ to zero:

$$\hat{\theta}_{\text{MAP}}(y) = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \bar{y} + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N} \mu_\theta \quad (4.67)$$

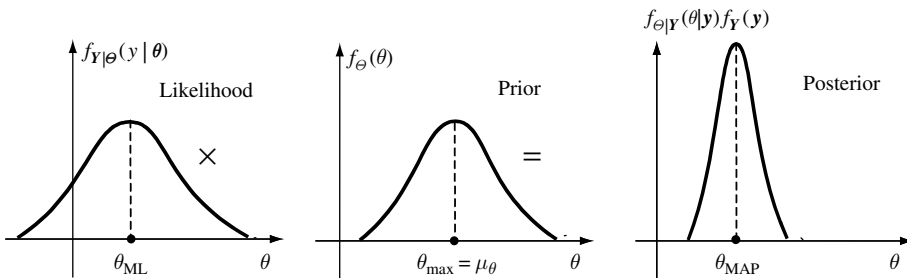


Figure 4.13 Illustration of the posterior pdf as product of the likelihood and the prior.

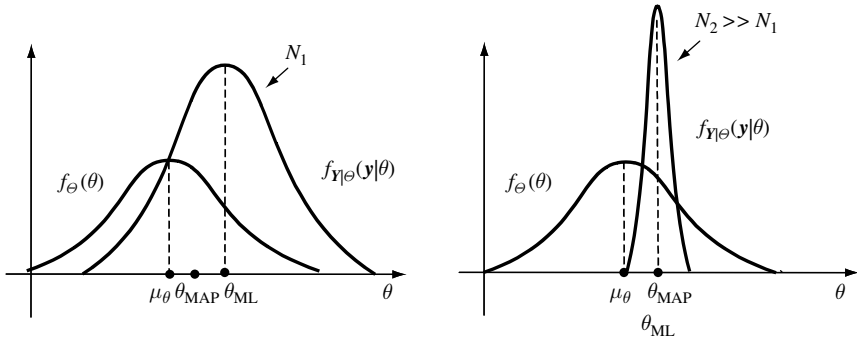


Figure 4.14 Illustration of the effect of increasing length of observation on the variance of an estimator.

where

$$\bar{y} = \sum_{m=0}^{N-1} y(m)/N$$

Note that the MAP estimate is an interpolation between the ML estimate \bar{y} and the mean of the prior pdf μ_θ , as shown in Figure 4.14. Also note that in Equation (4.67) the interpolation weights are dependent on signal-to-noise ratio and the length of the observation. As the variance (i.e. power) of noise decreases relative to the variance of the parameter and/or as the number of observations increases, the influence of the prior decreases, conversely, as the variance (i.e. power) of noise increases and/or as the number of observation decreases, the influence of the prior increases.

The expectation of the MAP estimate is obtained by noting that the only random variable on the right-hand side of Equation (4.67) is the term \bar{y} , and that $\mathcal{E}[\bar{y}] = \theta$

$$\mathcal{E}[\hat{\theta}_{\text{MAP}}(\mathbf{y})] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \theta + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N} \mu_\theta \tag{4.68}$$

and the variance of the MAP estimate is given as

$$\text{Var}[\hat{\theta}_{\text{MAP}}(\mathbf{y})] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \times \text{Var}[\bar{y}] = \frac{\sigma_n^2/N}{1 + \sigma_n^2/N\sigma_\theta^2} \tag{4.69}$$

Substitution of Equation (4.60) in Equation (4.69) yields

$$\text{Var}[\hat{\theta}_{\text{MAP}}(\mathbf{y})] = \frac{\text{Var}[\hat{\theta}_{\text{ML}}(\mathbf{y})]}{1 + \text{Var}[\hat{\theta}_{\text{ML}}(\mathbf{y})]/\sigma_\theta^2} \tag{4.70}$$

Note that, as σ_θ^2 , the variance of the parameter θ , increases, the influence of the prior decreases, and the variance of the MAP estimate tends towards the variance of the ML estimate.

4.2.7 THE RELATIVE IMPORTANCE OF THE PRIOR AND THE OBSERVATION

A fundamental issue in the Bayesian inference method is the relative influence of the observation signal and the prior pdf on the outcome. The importance of the observation depends on the confidence in the observation, and the confidence in turn depends on the length of the observation and on the signal-to-noise ratio (SNR). In general, as the number of observation samples and the SNR increase, the variance of the estimate and the influence of the prior decrease. From Equation (4.67), for the estimation of a Gaussian distributed parameter observed in AWGN, as the length of the observation N increases, the importance of the prior decreases, and the MAP estimate tends to the ML estimate:

$$\lim_{N \rightarrow \infty} \hat{\theta}_{\text{MAP}}(\mathbf{y}) = \lim_{N \rightarrow \infty} \left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \bar{y} + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N} \mu_\theta \right) = \bar{y} = \hat{\theta}_{\text{ML}} \quad (4.71)$$

As illustrated in Figure 4.14, as the length of the observation N tends to infinity, then both the MAP and the ML estimates of the parameter should tend to its true value, θ .

Example 4.9: MAP Estimation of a Signal in Additive Noise

Consider the estimation of a scalar-valued Gaussian signal $x(m)$, observed in an additive Gaussian white noise $n(m)$, and modelled as

$$y(m) = x(m) + n(m) \quad (4.72)$$

The posterior pdf of the signal $x(m)$ is given by

$$\begin{aligned} f_{X|Y}[x(m)|y(m)] &= \frac{1}{f_Y[y(m)]} f_{Y|X}[y(m)|x(m)] f_X[x(m)] \\ &= \frac{1}{f_Y[y(m)]} f_N[y(m) - x(m)] f_X[x(m)] \end{aligned} \quad (4.73)$$

where $f_X[x(m)] = \mathcal{N}[x(m), \mu_x, \sigma_x^2]$ and $f_N[n(m)] = \mathcal{N}[n(m), \mu_n, \sigma_n^2]$ are the Gaussian pdfs of the signal and noise, respectively. Substitution of the signal and noise pdfs in Equation (4.73) yields

$$\begin{aligned} f_{X|Y}[x(m)|y(m)] &= \frac{1}{f_Y[y(m)]} \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left\{ -\frac{[y(m) - x(m) - \mu_n]^2}{2\sigma_n^2} \right\} \\ &\quad \times \frac{1}{\sqrt{2\pi}\sigma_x} \exp \left\{ -\frac{[x(m) - \mu_x]^2}{2\sigma_x^2} \right\} \end{aligned} \quad (4.74)$$

This equation can be rewritten as

$$\begin{aligned} f_{X|Y}[x(m)|y(m)] &= \frac{1}{f_Y[y(m)]} \frac{1}{2\pi\sigma_n\sigma_x} \times \\ &\quad \exp \left\{ -\frac{\sigma_x^2 [y(m) - x(m) - \mu_n]^2 + \sigma_n^2 [x(m) - \mu_x]^2}{2\sigma_x^2\sigma_n^2} \right\} \end{aligned} \quad (4.75)$$

To obtain the MAP estimate we set the derivative of the log-likelihood function $\ln f_{x|Y}[x(m)|y(m)]$ with respect to $x(m)$ to zero as

$$\frac{\partial \{\ln f_{x|Y}[x(m)|y(m)]\}}{\partial \hat{x}(m)} = -\frac{-2\sigma_x^2 [y(m) - x(m) - \mu_n] + 2\sigma_n^2 [x(m) - \mu_x]}{2\sigma_x^2 \sigma_n^2} = 0 \quad (4.76)$$

From Equation (4.76) the MAP signal estimate is given by

$$\hat{x}(m) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2} [y(m) - \mu_n] + \frac{\sigma_n^2}{\sigma_x^2 + \sigma_n^2} \mu_x \quad (4.77)$$

Note that the estimate $\hat{x}(m)$ is a weighted linear interpolation between the unconditional mean of $x(m)$, μ_x , and the observed value $[y(m) - \mu_n]$. At a very poor SNR, i.e. when $\sigma_x^2 \ll \sigma_n^2$, we have $\hat{x}(m) \approx \mu_x$; on the other hand, for a noise-free signal $\sigma_n^2 = 0$ and $\mu_n = 0$, and we have $\hat{x}(m) = y(m)$.

Example 4.10: MAP Estimate of a Gaussian–AR Process Observed in AWGN

Consider a vector, \mathbf{x} , of N samples from an AR process observed in an additive Gaussian noise, and modelled as

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \quad (4.78)$$

From Chapter 8, a vector \mathbf{x} from an AR process may be expressed as

$$\mathbf{e} = \mathbf{A}\mathbf{x} \quad (4.79)$$

where \mathbf{A} is a matrix of the AR model coefficients, and the vector \mathbf{e} is the input signal of the AR model. Assuming that the signal \mathbf{x} is Gaussian, and that the P initial samples \mathbf{x}_0 are known, the pdf of the signal \mathbf{x} is given by

$$f_X(\mathbf{x}|\mathbf{x}_0) = f_E(\mathbf{e}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}\right) \quad (4.80)$$

where it is assumed that the input signal \mathbf{e} of the AR model is a zero-mean uncorrelated process with variance σ_e^2 . The pdf of a zero-mean Gaussian noise vector \mathbf{n} , with covariance matrix Σ_{nn} , is given by

$$f_N(\mathbf{n}) = \frac{1}{(2\pi)^{N/2} |\Sigma_{nn}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{n}^T \Sigma_{nn}^{-1} \mathbf{n}\right) \quad (4.81)$$

From Bayes' rule, the pdf of the signal given the noisy observation is

$$f_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{f_{Y|X}(\mathbf{y}|\mathbf{x}) f_X(\mathbf{x})}{f_Y(\mathbf{y})} = \frac{1}{f_Y(\mathbf{y})} f_N(\mathbf{y} - \mathbf{x}) f_X(\mathbf{x}) \quad (4.82)$$

Substitution of the pdfs of the signal and noise in Equation (4.82) yields

$$f_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{1}{f_Y(\mathbf{y}) (2\pi)^N \sigma_e^{N/2} |\Sigma_{nn}|^{1/2}} \exp \left\{ -\frac{1}{2} \left[(\mathbf{y} - \mathbf{x})^T \Sigma_{nn}^{-1} (\mathbf{y} - \mathbf{x}) + \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\sigma_e^2} \right] \right\} \quad (4.83)$$

The MAP estimate corresponds to the minimum of the argument of the exponential function in Equation (4.83). Assuming that the argument of the exponential function is differentiable, and has a well-defined minimum, we can obtain the MAP estimate from

$$\hat{\mathbf{x}}_{\text{MAP}}(\mathbf{y}) = \arg \operatorname{zero}_x \left\{ \frac{\partial}{\partial \mathbf{x}} \left[(\mathbf{y} - \mathbf{x})^T \Sigma_{nn}^{-1} (\mathbf{y} - \mathbf{x}) + \frac{\mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}{\sigma_e^2} \right] \right\} \quad (4.84)$$

The MAP estimate is

$$\hat{\mathbf{x}}_{\text{MAP}}(\mathbf{y}) = \left(\mathbf{I} + \frac{1}{\sigma_e^2} \Sigma_{nn} \mathbf{A}^T \mathbf{A} \right)^{-1} \mathbf{y} \quad (4.85)$$

where \mathbf{I} is the identity matrix.

4.3 THE ESTIMATE-MAXIMISE METHOD

The EM algorithm is an iterative likelihood maximisation method with applications in blind de-convolution, clustering, training of hidden Markov models, model-based signal interpolation, spectral estimation from noisy observations, signal restoration and estimation of a set of model parameters from a training data set.

The EM is a framework for solving problems where it is difficult to obtain a direct ML estimate, either because the data is incomplete (Figure 4.15), e.g. when there are missing samples or missing labels, or because the problem is difficult. For example, in clustering applications usually the raw data do not have a cluster label attached to them and hence an iterative EM process is employed consisting of (a) labelling of data (expectation) and (b) calculation of means and variances of clusters.

To define the term *incomplete data*, consider a signal \mathbf{x} from a random process X with an unknown parameter vector, $\boldsymbol{\theta}$, and a pdf $f_{X;\boldsymbol{\theta}}(\mathbf{x}; \boldsymbol{\theta})$. The notation $f_{X;\boldsymbol{\theta}}(\mathbf{x}; \boldsymbol{\theta})$ expresses the dependence of the pdf of X on the value of the unknown parameter $\boldsymbol{\theta}$. The signal \mathbf{x} is the so-called *complete data* and the ML estimate of the parameter vector $\boldsymbol{\theta}$ may be obtained from $f_{X;\boldsymbol{\theta}}(\mathbf{x}; \boldsymbol{\theta})$. Now assume that the signal \mathbf{x} goes through a many-to-one noninvertible transformation (e.g. when a number of samples of the vector \mathbf{x} are lost) and is observed as \mathbf{y} . The observation \mathbf{y} is the so-called ‘incomplete data’.

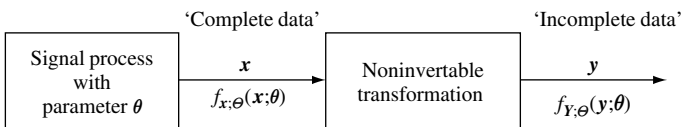


Figure 4.15 Illustration of transformation of complete data to incomplete data.

Maximisation of the likelihood of the incomplete data, $f_{Y;\theta}(\mathbf{y}; \theta)$, with respect to the parameter vector, θ , is often a difficult task, whereas maximisation of the likelihood of the complete data, $f_{X;\theta}(\mathbf{x}; \theta)$, is relatively easy. Since the complete data is unavailable, the parameter estimate is obtained through maximisation of the *conditional expectation* of the log-likelihood of the complete data defined as

$$\mathcal{E} [\ln f_{X;\theta}(\mathbf{x}; \theta) | \mathbf{y}] = \int_{\mathbf{x}} f_{X|Y;\theta}(\mathbf{x} | \mathbf{y}; \theta) \ln f_{X;\theta}(\mathbf{x}; \theta) d\mathbf{x} \quad (4.86)$$

In Equation (4.86), the computation of the term $f_{X|Y;\theta}(\mathbf{x} | \mathbf{y}; \theta)$ requires an estimate of the unknown parameter vector, θ . For this reason, the expectation of the likelihood function is maximised iteratively starting with an initial estimate of θ , and updating the estimate as described below. Note that the right-hand side of Equation (4.86) is similar to an entropy function.

EM Algorithm

Step 1: initialisation – select an initial parameter estimate θ_0 , and for $i = 0, 1, \dots$ until convergence:

Step 2: expectation – compute

$$\begin{aligned} U(\theta, \hat{\theta}_i) &= E[\ln f_{X;\theta}(\mathbf{x}; \theta) | \mathbf{y}; \hat{\theta}_i] \\ &= \int_{\mathbf{x}} f_{X|Y;\theta}(\mathbf{x} | \mathbf{y}; \hat{\theta}_i) \ln f_{X;\theta}(\mathbf{x}; \theta) d\mathbf{x} \end{aligned} \quad (4.87)$$

Step 3: maximisation – select

$$\hat{\theta}_{i+1} = \arg \max_{\theta} U(\theta, \hat{\theta}_i) \quad (4.88)$$

Step 4: convergence test – if not converged then go to Step 2.

4.3.1 CONVERGENCE OF THE EM ALGORITHM

In this section, it is shown that the EM algorithm converges to a maximum of the likelihood of the incomplete data, $f_{Y;\theta}(\mathbf{y}; \theta)$. The likelihood of the complete data can be written as

$$f_{X,Y;\theta}(\mathbf{x}, \mathbf{y}; \theta) = f_{X|Y;\theta}(\mathbf{x} | \mathbf{y}; \theta) f_{Y;\theta}(\mathbf{y}; \theta) \quad (4.89)$$

where $f_{X,Y;\theta}(\mathbf{x}, \mathbf{y}; \theta)$ is the likelihood of \mathbf{x} and \mathbf{y} with θ as a parameter. From Equation (4.89), the log-likelihood of the incomplete data is obtained as

$$\ln f_{Y;\theta}(\mathbf{y}; \theta) = \ln f_{X,Y;\theta}(\mathbf{x}, \mathbf{y}; \theta) - \ln f_{X|Y;\theta}(\mathbf{x} | \mathbf{y}; \theta) \quad (4.90)$$

Using an estimate, $\hat{\boldsymbol{\theta}}_i$, of the parameter vector, $\boldsymbol{\theta}$, and taking the expectation of Equation (4.90) over the space of the complete signal \mathbf{x} , we obtain

$$\ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \boldsymbol{\theta}) = U(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_i) - V(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_i) \quad (4.91)$$

where for a given \mathbf{y} , $\mathcal{E}[\ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \boldsymbol{\theta})] = \ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \boldsymbol{\theta})$, and the function $U(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_i)$ is the conditional expectation of $\ln f_{X,Y;\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$:

$$\begin{aligned} U(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) &= \mathcal{E}[\ln f_{X,Y;\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) | \mathbf{y}; \hat{\boldsymbol{\theta}}_i] \\ &= \int_X f_{X|Y;\boldsymbol{\theta}}(\mathbf{x} | \mathbf{y}; \hat{\boldsymbol{\theta}}_i) \ln f_{X;\boldsymbol{\theta}}(\mathbf{x}; \boldsymbol{\theta}) \, d\mathbf{x} \end{aligned} \quad (4.92)$$

The function $V(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ is the conditional expectation of $\ln f_{X|Y;\boldsymbol{\theta}}(\mathbf{x} | \mathbf{y}; \boldsymbol{\theta})$:

$$\begin{aligned} V(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) &= \mathcal{E} \left[\ln f_{X|Y;\boldsymbol{\theta}}(\mathbf{x} | \mathbf{y}; \boldsymbol{\theta}) \mid \mathbf{y}; \hat{\boldsymbol{\theta}}_i \right] \\ &= \int_X f_{X|Y;\boldsymbol{\theta}}(\mathbf{x} | \mathbf{y}; \hat{\boldsymbol{\theta}}_i) \ln f_{X|Y;\boldsymbol{\theta}}(\mathbf{x} | \mathbf{y}; \boldsymbol{\theta}) \, d\mathbf{x} \end{aligned} \quad (4.93)$$

Now, from Equation (4.91), the log-likelihood of the incomplete data \mathbf{y} with parameter estimate $\hat{\boldsymbol{\theta}}_i$ at iteration i is

$$\ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \hat{\boldsymbol{\theta}}_i) = U(\hat{\boldsymbol{\theta}}_i; \hat{\boldsymbol{\theta}}_i) - V(\hat{\boldsymbol{\theta}}_i; \hat{\boldsymbol{\theta}}_i) \quad (4.94)$$

It can be shown (see Dempster *et al.*, 1977) that the function V satisfies the inequality

$$V(\hat{\boldsymbol{\theta}}_{i+1}; \hat{\boldsymbol{\theta}}_i) \leq V(\hat{\boldsymbol{\theta}}_i; \hat{\boldsymbol{\theta}}_i) \quad (4.95)$$

and in the maximisation step of EM we choose $\hat{\boldsymbol{\theta}}_{i+1}$ such that

$$U(\hat{\boldsymbol{\theta}}_{i+1}; \hat{\boldsymbol{\theta}}_i) \geq U(\hat{\boldsymbol{\theta}}_i; \hat{\boldsymbol{\theta}}_i) \quad (4.96)$$

From Equation (4.94) and the inequalities (4.95) and (4.96), it follows that

$$\ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \hat{\boldsymbol{\theta}}_{i+1}) \geq \ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \hat{\boldsymbol{\theta}}_i) \quad (4.97)$$

Therefore at every iteration of the EM algorithm, the conditional likelihood of the estimate increases until the estimate converges to a local maximum of the log-likelihood function, $\ln f_{Y;\boldsymbol{\theta}}(\mathbf{y}; \boldsymbol{\theta})$.

The EM algorithm is applied to the solution of a number of problems in this book. In Section 4.5, the estimation of the parameters of a mixture Gaussian model for the signal space of a recorded process is formulated in an EM framework. In Chapter 5, the EM is used for estimation of the parameters of a hidden Markov model.

4.4 CRAMER–RAO BOUND ON THE MINIMUM ESTIMATOR VARIANCE

An important measure of the performance of an estimator is the variance of the estimate with the varying values of the observation signal, \mathbf{y} , and the parameter vector, $\boldsymbol{\theta}$. The minimum estimation variance depends on the distributions of the parameter vector, $\boldsymbol{\theta}$, and on the observation signal, \mathbf{y} . In this section, we first consider the lower bound on the variance of the estimates of a constant parameter, and then extend the results to random parameters.

The Cramer–Rao lower bound on the variance of estimate of the i th coefficient, θ_i , of a parameter vector, $\boldsymbol{\theta}$, is given as

$$\text{Var}[\hat{\theta}_i(\mathbf{y})] \geq \frac{\left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i}\right)^2}{\mathcal{E}\left[\left(\frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right]} \tag{4.98}$$

An estimator that achieves the lower bound on the variance is called the minimum variance, or the most efficient, estimator.

Proof The bias in the estimate $\hat{\theta}_i(\mathbf{y})$ of the i th coefficient of the parameter vector $\boldsymbol{\theta}$, averaged over the observation space \mathbf{Y} , is defined as

$$\mathcal{E}[\hat{\theta}_i(\mathbf{y}) - \theta_i] = \int_{-\infty}^{\infty} [\hat{\theta}_i(\mathbf{y}) - \theta_i] f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \, d\mathbf{y} = \theta_{\text{Bias}} \tag{4.99}$$

Differentiation of Equation (4.99) with respect to θ_i yields

$$\int_{-\infty}^{\infty} \left\{ [\hat{\theta}_i(\mathbf{y}) - \theta_i] \frac{\partial f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i} - f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \right\} \, d\mathbf{y} = \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \tag{4.100}$$

For a probability density function we have

$$\int_{-\infty}^{\infty} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \, d\mathbf{y} = 1 \tag{4.101}$$

Therefore Equation (4.100) can be written as

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\mathbf{y}) - \theta_i] \frac{\partial f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i} \, d\mathbf{y} = 1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \tag{4.102}$$

Now, since the derivative of the integral of a pdf is zero, taking the derivative of Equation (4.101) and multiplying the result by θ_{Bias} yields

$$\theta_{\text{Bias}} \int_{-\infty}^{\infty} \frac{\partial f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i} \, d\mathbf{y} = 0 \tag{4.103}$$

Substituting

$$\partial f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})/\partial\theta_i = f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})/\partial\theta_i$$

into Equation (4.102), and using Equation (4.103), we obtain

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\mathbf{y}) - \theta_{\text{Bias}} - \theta_i] \frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \, d\mathbf{y} = 1 + \frac{\partial\theta_{\text{Bias}}}{\partial\theta_i} \quad (4.104)$$

Now squaring both sides of Equation (4.104), we obtain

$$\left(\int_{-\infty}^{\infty} [\hat{\theta}_i(\mathbf{y}) - \theta_{\text{Bias}} - \theta_i] \frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \, d\mathbf{y} \right)^2 = \left(1 + \frac{\partial\theta_{\text{Bias}}}{\partial\theta_i} \right)^2 \quad (4.105)$$

For the left-hand side of Equation (4.105) application of the following Schwartz inequality

$$\left(\int_{-\infty}^{\infty} f(y)g(y) \, dy \right)^2 \leq \int_{-\infty}^{\infty} [f(y)]^2 \, dy \times \int_{-\infty}^{\infty} [g(y)]^2 \, dy \quad (4.106)$$

yields

$$\begin{aligned} & \left\{ \int_{-\infty}^{\infty} ([\hat{\theta}_i(\mathbf{y}) - \theta_{\text{Bias}} - \theta_i] f_{Y|\boldsymbol{\theta}}^{1/2}(\mathbf{y}|\boldsymbol{\theta})) \left(\frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} f_{Y|\boldsymbol{\theta}}^{1/2}(\mathbf{y}|\boldsymbol{\theta}) \right) \, d\mathbf{y} \right\}^2 \\ & \leq \left\{ \left(\int_{-\infty}^{\infty} ([\hat{\theta}_i(\mathbf{y}) - \theta_{\text{Bias}} - \theta_i]^2 f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})) \, d\mathbf{y} \right) \right\} \left\{ \int_{-\infty}^{\infty} \left(\frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} \right)^2 f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}) \, d\mathbf{y} \right\} \end{aligned} \quad (4.107)$$

From Equations (4.105) and (4.107), we have

$$\text{Var}[\hat{\theta}_i(\mathbf{y})] \times \mathcal{E} \left[\left(\frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} \right)^2 \right] \geq \left(1 + \frac{\partial\theta_{\text{Bias}}}{\partial\theta_i} \right)^2 \quad (4.108)$$

The Cramer–Rao inequality (4.98) results directly from the inequality (4.108).

4.4.1 CRAMER–RAO BOUND FOR RANDOM PARAMETERS

For random parameters the Cramer–Rao bound may be obtained using the same procedure as above, with the difference that, in Equation (4.98), instead of the likelihood $f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})$ we use the joint pdf, $f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})$, and we also use the logarithmic relation

$$\frac{\partial \ln f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})}{\partial\theta_i} = \frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial\theta_i} + \frac{\partial \ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{\partial\theta_i} \quad (4.109)$$

The Cramer–Rao bound for random parameters is obtained as

$$\text{Var}[\hat{\theta}_i(\mathbf{y})] \geq \frac{\left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i}\right)^2}{\mathcal{E} \left[\left(\frac{\partial \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i} \right)^2 + \left(\frac{\partial \ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{\partial \theta_i} \right)^2 \right]} \quad (4.110)$$

where the second term in the denominator of Equation (4.110) describes the effect of the prior pdf of $\boldsymbol{\theta}$. As expected, the use of the prior, $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, can result in a decrease in the variance of the estimate. An alternative form of the minimum bound on estimation variance can be obtained using the likelihood relation

$$\mathcal{E} \left[\left(\frac{\partial \ln f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})}{\partial \theta_i} \right)^2 \right] = -\mathcal{E} \left[\frac{\partial^2 \ln f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})}{\partial \theta_i^2} \right] \quad (4.111)$$

as

$$\text{Var}[\hat{\theta}_i(\mathbf{y})] \geq -\frac{\left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i}\right)^2}{\mathcal{E} \left[\frac{\partial^2 \ln f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta})}{\partial \theta_i^2} + \frac{\partial^2 \ln f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{\partial \theta_i^2} \right]} \quad (4.112)$$

4.4.2 CRAMER–RAO BOUND FOR A VECTOR PARAMETER

For real-valued P -dimensional vector parameters, the Cramer–Rao bound for the covariance matrix of an unbiased estimator of $\boldsymbol{\theta}$ is given by

$$\text{Cov}[\hat{\boldsymbol{\theta}}] \geq \mathbf{J}^{-1}(\boldsymbol{\theta}) \quad (4.113)$$

where \mathbf{J} is the $P \times P$ Fisher information matrix, with elements given by

$$[\mathbf{J}(\boldsymbol{\theta})]_{ij} = -\mathcal{E} \left[\frac{\partial^2 \ln f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right] \quad (4.114)$$

The lower bound on the variance of the i th element of the vector $\boldsymbol{\theta}$ is given by

$$\text{Var}(\hat{\theta}_i) \geq [\mathbf{J}^{-1}(\boldsymbol{\theta})]_{ii} = \frac{1}{\mathcal{E} \left[\frac{\partial^2 \ln f_{Y,\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\theta})}{\partial \theta_i^2} \right]} \quad (4.115)$$

where $[\mathbf{J}^{-1}(\boldsymbol{\theta})]_{ii}$ is the i th diagonal element of the inverse of the Fisher matrix.

4.5 DESIGN OF GAUSSIAN MIXTURE MODELS

A practical method for modelling the probability density function of an arbitrary signal space is to fit (or ‘tile’) the space with a mixture of a number of Gaussian probability density

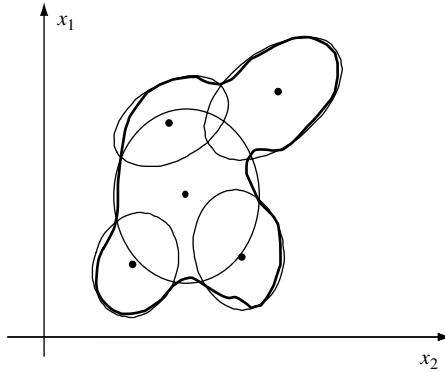


Figure 4.16 Illustration of probabilistic modelling of a two-dimensional signal space with a mixture of five bivariate Gaussian densities.

functions. The Gaussian functions hence act as elementary pdfs from which other pdfs can be constructed.

Figure 4.16 illustrates the cluster modelling of a two-dimensional signal space with a number of circular and elliptically shaped Gaussian processes. Note that the Gaussian densities can be overlapping, with the result that, in an area of overlap, a data point can be associated with different components of the Gaussian mixture.

A main advantage of the use of a mixture Gaussian model is that it results in mathematically tractable signal processing solutions. A mixture Gaussian pdf model for a process X is defined as

$$f_X(\mathbf{x}) = \sum_{k=1}^K P_k \mathcal{N}_k(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (4.116)$$

where $\mathcal{N}_k(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the k th component of the mixture Gaussian pdf, with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The parameter P_k is the prior probability of the k th mixture, and it can be interpreted as the expected fraction of the number of vectors from the process X associated with the k th mixture.

In general, there are an infinite number of different K -mixture Gaussian densities that can be used to ‘tile up’ a signal space. Hence the modelling of a signal space with a K -mixture pdf space can be regarded as a many-to-one mapping, and the EM method can be applied for the estimation of the parameters of the Gaussian pdf models.

4.5.1 EM ESTIMATION OF GAUSSIAN MIXTURE MODEL

The EM algorithm, discussed in Section 4.4, is an iterative ML estimation method, and can be employed to calculate the parameters of a K -mixture Gaussian pdf model for a given data set. To apply the EM method we first need to define the so-called complete and incomplete data sets. As usual, the observation vectors $[\mathbf{y}(m) \ m = 0, \dots, N - 1]$ form the incomplete data. The complete data may be viewed as the observation vectors with a *label* attached to each vector $\mathbf{y}(m)$ to indicate the component of the mixture Gaussian model that generated the vector. Note that, if each signal vector $\mathbf{y}(m)$ had a mixture component label attached,

then the computation of the mean vector and the covariance matrix of each component of the mixture would be a relatively simple exercise. Therefore the complete and incomplete data can be defined as follows:

The *incomplete* data $\mathbf{y}(m)$, $m = 0, \dots, N-1$

The *complete* data $\mathbf{x}(m) = [\mathbf{y}(m), k] = \mathbf{y}_k(m)$, $m = 0, \dots, N-1$, $k \in (1, \dots, K)$

The probability of the complete data is the probability that an observation vector $\mathbf{y}(m)$ has a label k associating it with the k th component of the mixture density. The main step in application of the EM method is to define the expectation of the complete data, given the observations and a current estimate of the parameter vector, as

$$\begin{aligned} U(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}_i) &= \mathcal{E}[\ln f_{Y,K;\boldsymbol{\Theta}}[\mathbf{y}(m), k; \boldsymbol{\Theta}] | \mathbf{y}(m); \hat{\boldsymbol{\Theta}}_i] \\ &= \sum_{m=0}^{N-1} \sum_{k=1}^K \frac{f_{Y,K|\boldsymbol{\Theta}}[\mathbf{y}(m), k | \hat{\boldsymbol{\Theta}}_i]}{f_{Y|\boldsymbol{\Theta}}[\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i]} \ln f_{Y,K;\boldsymbol{\Theta}}[\mathbf{y}(m), k; \boldsymbol{\Theta}] \end{aligned} \quad (4.117)$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k = [P_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k], k = 1, \dots, K\}$, are the parameters of the Gaussian mixture as in Equation (4.116). Now the joint pdf of $\mathbf{y}(m)$ and the k th Gaussian component of the mixture density can be written as

$$\begin{aligned} f_{Y,K|\boldsymbol{\Theta}}[\mathbf{y}(m), k | \hat{\boldsymbol{\theta}}_i] &= P_{k_i} f_k[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_{k_i}] \\ &= P_{k_i} \mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}] \end{aligned} \quad (4.118)$$

where $\mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]$ is a Gaussian density with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$:

$$\mathcal{N}_k[\mathbf{y}(m); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k] = \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp \left\{ -\frac{1}{2} [\mathbf{y}(m) - \boldsymbol{\mu}_k]^T \boldsymbol{\Sigma}_k^{-1} [\mathbf{y}(m) - \boldsymbol{\mu}_k] \right\} \quad (4.119)$$

The pdf of $\mathbf{y}(m)$ as a mixture of K Gaussian densities is given by

$$\begin{aligned} f_{Y|\boldsymbol{\Theta}}[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_i] &= \mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_i] \\ &= \sum_{k=1}^K \hat{P}_{k_i} \mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}] \end{aligned} \quad (4.120)$$

Substitution of the Gaussian densities of Equation (4.118) and Equation (4.120) in Equation (4.117) yields

$$\begin{aligned} U[(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\mathbf{P}}_i)] &= \sum_{m=0}^{N-1} \sum_{k=1}^K \frac{\hat{P}_{k_i} \mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_i]} \ln \{P_k \mathcal{N}_k[\mathbf{y}(m); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k]\} \\ &= \sum_{m=0}^{N-1} \sum_{k=1}^K \left\{ \frac{\hat{P}_{k_i} \mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_i]} \ln P_{k_i} \right. \\ &\quad \left. + \frac{\hat{P}_{k_i} \mathcal{N}_k[\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\theta}}_i]} \ln \mathcal{N}[\mathbf{y}(m); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k] \right\} \end{aligned} \quad (4.121)$$

Equation (4.121) is maximised with respect to the parameter P_k using the constrained optimisation method. This involves subtracting the constant term $\sum P_k = 1$ from the right-hand side of Equation (4.121) and then setting the derivative of this equation with respect to P_k to zero; this yields

$$\hat{P}_{k_{i+1}} = \arg \max_{P_k} U [(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \mathbf{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{P}_i)] = \frac{1}{N} \sum_{m=0}^{N-1} \frac{\hat{P}_{k_i} \mathcal{N}_k [\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}(\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i)} \quad (4.122)$$

The parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ that maximise the function U are obtained by setting the derivative of the function with respect to these parameters to zero:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{k_{i+1}} &= \arg \max_{\hat{\boldsymbol{\mu}}_k} U [(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \mathbf{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{P}_i)] \\ &= \frac{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i} \mathcal{N}_k [\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i]} \mathbf{y}(m)}{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i} \mathcal{N}_k [\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i]}} \end{aligned} \quad (4.123)$$

and

$$\begin{aligned} \hat{\boldsymbol{\Sigma}}_{k_{i+1}} &= \arg \max_{\hat{\boldsymbol{\Sigma}}_k} U [(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \mathbf{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{P}_i)] \\ &= \frac{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i} \mathcal{N}_k [\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i]} [\mathbf{y}(m) - \hat{\boldsymbol{\mu}}_{k_i}] [\mathbf{y}(m) - \hat{\boldsymbol{\mu}}_{k_i}]^T}{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i} \mathcal{N}_k [\mathbf{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}]}{\mathcal{N}[\mathbf{y}(m) | \hat{\boldsymbol{\Theta}}_i]}} \end{aligned} \quad (4.124)$$

Equations (4.122)–(4.124) are the estimates of the parameters of a mixture Gaussian pdf model. These equations can be used in further iterations of the EM method until the parameter estimates converge.

4.6 BAYESIAN CLASSIFICATION

Classification is the process of *labelling* of an observation sequence $\{\mathbf{y}(m)\}$ with one of M classes of signals $\{C_k; k = 1, \dots, M\}$ that could have generated the observation. Classifiers are present in all modern digital communication systems and in applications such as the decoding of discrete-valued symbols in digital communication receivers, speech compression, video compression, speech recognition, image recognition, character recognition, signal/noise classification and detectors.

For example, in an M -symbol digital communication system, the channel output signal is classified as one of the M signalling symbols; in speech recognition, segments of speech

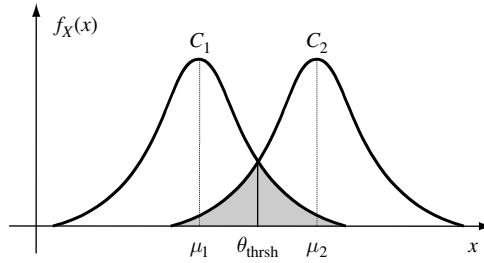


Figure 4.17 Illustration of the overlap of the distribution of two classes of signals.

signals are labelled with one of about 40 elementary phonemes sounds; and in speech or video compression, a segment of speech samples or a block of image pixels is quantised and labelled with one of a number of prototype signal vectors in a codebook. In the design of a classifier, the aim is to reduce the classification error given the constraints on the signal-to-noise ratio, available training data, bandwidth and the computational resources.

Classification errors are due to overlap of the distributions of different classes of signals. This is illustrated in Figure 4.17 for a binary classification problem with two Gaussian distributed signal classes C_1 and C_2 . In the shaded region, where the signal distributions overlap, a sample x could belong to either of the two classes. The shaded area gives a measure of the classification error. The obvious solution suggested by Figure 4.17 for reducing the classification error is to reduce the overlap of the distributions. The overlap can be reduced in two ways: (a) by increasing the distance between the mean values of different classes; and (b) by reducing the variance of each class. In telecommunications systems the overlap between the signal classes is reduced using a combination of several methods including increasing the signal-to-noise ratio, increasing the distance between signal patterns by adding redundant error control coding bits, and signal shaping and post-filtering operations. In pattern recognition, where it is not possible to control the signal generation process (as in speech and image recognition), the choice of the pattern features and models affects the classification error. The design of an efficient classification for pattern recognition depends on a number of factors, which can be listed as follows:

- (1) Extraction and transformation of a set of discriminative features from the signal that can aid the classification process. The features need to adequately characterise each class and emphasise the differences between various classes.
- (2) Statistical modelling of the observation features for each class. For Bayesian classification, a posterior probability model for each class should be obtained.
- (3) Labelling of an unlabelled signal with one of the N classes.

4.6.1 BINARY CLASSIFICATION

The simplest form of classification is the labelling of an observation with one of two classes of signals. Figures 4.18(a) and (b) illustrate two examples of a simple binary classification problem in a two-dimensional signal space. In each case, the observation is the result of a

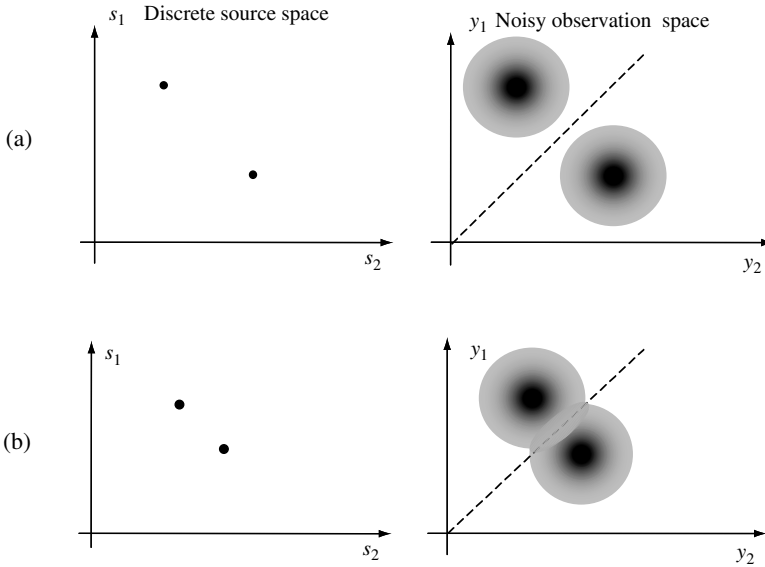


Figure 4.18 Illustration of binary classification: (a) the source and observation spaces are well separated; (b) the observation spaces overlap.

random mapping (e.g. signal plus noise) from the binary source to the continuous observation space. In Figure 4.18(a), the binary sources and the observation space associated with each source are well separated, and it is possible to make an error-free classification of each observation. In Figure 4.18(b) there is less distance between the mean of the sources, and the observation signals have a greater spread. These result in some overlap of the signal spaces and classification error can occur.

In binary classification, a signal \mathbf{x} is labelled with the class that scores the higher *a posteriori* probability:

$$P_{C|\mathbf{x}}(C_1|\mathbf{x}) \geq P_{C|\mathbf{x}}(C_2|\mathbf{x}) \tag{4.125}$$

Note the above notation means that a signal \mathbf{x} is classified as C_1 if $P_{C|\mathbf{x}}(C_1|\mathbf{x}) > P_{C|\mathbf{x}}(C_2|\mathbf{x})$, otherwise it is classified as C_2 . Using Bayes' rule, Equation (4.125) can be rewritten as

$$P_C(C_1) f_{\mathbf{x}|C}(\mathbf{x}|C_1) \geq P_C(C_2) f_{\mathbf{x}|C}(\mathbf{x}|C_2) \tag{4.126}$$

Letting $P_C(C_1) = P_1$ and $P_C(C_2) = P_2$, Equation (4.126) is often written in terms of a *likelihood ratio test* as

$$\frac{f_{\mathbf{x}|C}(\mathbf{x}|C_1)}{f_{\mathbf{x}|C}(\mathbf{x}|C_2)} \geq \frac{P_2}{P_1} \tag{4.127}$$

Taking the likelihood ratio yields the following discriminant function:

$$h(\mathbf{x}) = -\ln f_{\mathbf{x}|C_1}(\mathbf{x}|C_1) - \ln f_{\mathbf{x}|C_2}(\mathbf{x}|C_2) \underset{C_2}{\overset{C_1}{\geq}} \ln \frac{P_1}{P_2} \quad (4.128)$$

Now assume that the signal in each class has a Gaussian distribution with a probability distribution function given by

$$f_{\mathbf{x}|C_i}(\mathbf{x}|c_i) = \frac{1}{\sqrt{2\pi}|\Sigma_i|} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right], i = 1, 2 \quad (4.129)$$

From Equations (4.128) and (4.129), the discriminant function $h(\mathbf{x})$ becomes

$$h(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\mathbf{x} - \boldsymbol{\mu}_2) + \ln \frac{|\Sigma_2|}{|\Sigma_1|} \underset{C_2}{\overset{C_1}{\geq}} \ln \frac{P_2}{P_1} \quad (4.130)$$

Example 4.11

For two Gaussian-distributed classes of scalar-valued signals with distributions given by $\mathcal{N}[x(m), \mu_1, \sigma^2]$ and $\mathcal{N}[x(m), \mu_2, \sigma^2]$, and equal class probability, $P_1 = P_2 = 0.5$, the discrimination function of Equation (4.130) becomes

$$h[x(m)] = \frac{\mu_2 - \mu_1}{\sigma^2} x(m) + \frac{1}{2} \frac{\mu_2^2 - \mu_1^2}{\sigma^2} \underset{C_2}{\overset{C_1}{\geq}} 0 \quad (4.131)$$

Hence the rule for signal classification becomes

$$x(m) \underset{C_2}{\overset{C_1}{\leq}} \frac{\mu_1 + \mu_2}{2} \quad (4.132)$$

The signal is labelled with class C_1 if $x(m) < (\mu_1 + \mu_2)/2$ and as class C_2 otherwise.

4.6.2 CLASSIFICATION ERROR

Classification errors are due to the overlap of the distributions of different classes of signals. This is illustrated in Figure 4.17 for the binary classification of a scalar-valued signal and in Figure 4.18 for the binary classification of a two-dimensional signal. In each figure the overlapped area gives a measure of the classification error. The obvious solution for reducing the classification error is to reduce the overlap of the distributions. This may be achieved by increasing the distance between the mean values of various classes or by reducing the variance of each class. In the binary classification of a scalar-valued variable x , the probability of classification error is given by

$$P(\text{Error} | x) = P(C_1) P(x > \text{Thresh} | x \in C_1) + P(C_2) P(x < \text{Thresh} | x \in C_2) \quad (4.133)$$

For two Gaussian-distributed classes of scalar-valued signals with pdfs,

$$\mathcal{N}[x(m), \mu_1, \sigma_1^2] \quad \text{and} \quad \mathcal{N}[x(m), \mu_2, \sigma_2^2]$$

Equation (4.133) becomes

$$\begin{aligned} P(\text{Error}|x) = & P(C_1) \int_{\text{Thrsh}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right] dx \\ & + P(C_2) \int_{-\infty}^{\text{Thrsh}} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right] dx \end{aligned} \quad (4.134)$$

where the parameter Thrsh is the classification threshold.

4.6.3 BAYESIAN CLASSIFICATION OF DISCRETE-VALUED PARAMETERS

Let the set $\Theta = \{\theta_i, i = 1, \dots, M\}$ denote the values that a discrete P -dimensional parameter vector θ can assume. In general, the observation space \mathbf{Y} associated with a discrete parameter space Θ may be a discrete-valued space or a continuous-valued space; an example of the latter is a discrete-valued parameter observed in continuous-valued noise.

Assuming the observation space is continuous, the pdf of the parameter vector, θ_i , given observation \mathbf{y} , may be expressed using Bayes's rule as

$$P_{\theta_i|\mathbf{y}}(\theta_i|\mathbf{y}) = \frac{f_{\mathbf{y}|\theta}(\mathbf{y}|\theta_i)P_{\theta}(\theta_i)}{f_{\mathbf{y}}(\mathbf{y})} \quad (4.135)$$

For the case when the observation space \mathbf{Y} is discrete-valued, the probability density functions are replaced by the appropriate probability mass functions. The Bayesian risk in selecting the parameter vector θ_i given the observation \mathbf{y} is defined as

$$\mathcal{R}(\theta_i|\mathbf{y}) = \sum_{j=1}^M C(\theta_i|\theta_j)P_{\theta|\mathbf{y}}(\theta_j|\mathbf{y}) \quad (4.136)$$

where $C(\theta_i|\theta_j)$ is the cost of selecting the parameter θ_i when the true parameter is θ_j . The Bayesian classification, Equation (4.136), can be employed to obtain the MAP, ML and MMSE classifiers as described next.

4.6.4 MAXIMUM A POSTERIORI CLASSIFICATION

MAP classification corresponds to Bayesian classification with a uniform cost function defined as

$$C(\theta_i|\theta_j) = 1 - \delta(\theta_i, \theta_j) \quad (4.137)$$

where $\delta(\cdot)$ is the delta function. Substitution of this cost function in the Bayesian risk function yields

$$\begin{aligned}\mathcal{R}_{\text{MAP}}(\boldsymbol{\theta}_i|\mathbf{y}) &= \sum_{j=1}^M [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)] P_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}_j|\mathbf{y}) \\ &= 1 - P_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}_i|\mathbf{y})\end{aligned}\quad (4.138)$$

Note that the MAP risk in selecting $\boldsymbol{\theta}_i$ is the classification error probability, that is the sum of the probabilities of all other candidates. From Equation (4.138), minimisation of the MAP risk function is achieved by maximisation of the posterior pmf:

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\text{MAP}}(\mathbf{y}) &= \arg \max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}_i|\mathbf{y}) \\ &= \arg \max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\theta}}(\boldsymbol{\theta}_i) f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}_i)\end{aligned}\quad (4.139)$$

4.6.5 MAXIMUM-LIKELIHOOD CLASSIFICATION

The ML classification corresponds to Bayesian classification when the parameter $\boldsymbol{\theta}$ has a uniform prior pmf and the cost function is also uniform:

$$\begin{aligned}\mathcal{R}_{\text{ML}}(\boldsymbol{\theta}_i|\mathbf{y}) &= \sum_{j=1}^M [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)] \frac{1}{f_Y(\mathbf{y})} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}_j) P_{\boldsymbol{\theta}}(\boldsymbol{\theta}_j) \\ &= 1 - \frac{1}{f_Y(\mathbf{y})} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}_i) P_{\boldsymbol{\theta}}\end{aligned}\quad (4.140)$$

where $P_{\boldsymbol{\theta}}$ is the uniform pmf of $\boldsymbol{\theta}$. Minimisation of the ML risk function, Equation (4.140), is equivalent to maximisation of the likelihood $f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}_i)$,

$$\hat{\boldsymbol{\theta}}_{\text{ML}}(\mathbf{y}) = \arg \max_{\boldsymbol{\theta}_i} f_{Y|\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{\theta}_i)\quad (4.141)$$

4.6.6 MINIMUM MEAN SQUARE ERROR CLASSIFICATION

The Bayesian minimum mean square error classification results from minimisation of the following risk function:

$$\mathcal{R}_{\text{MMSE}}(\boldsymbol{\theta}_i|\mathbf{y}) = \sum_{j=1}^M |\boldsymbol{\theta}_i - \boldsymbol{\theta}_j|^2 P_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}_j|\mathbf{y})\quad (4.142)$$

For the case when $P_{(\boldsymbol{\theta}|\mathbf{y})}(\boldsymbol{\theta}|\mathbf{y})$ is not available, the MMSE classifier is given by

$$\hat{\boldsymbol{\theta}}_{\text{MMSE}}(\mathbf{y}) = \arg \min_{\boldsymbol{\theta}_i} |\boldsymbol{\theta}_i - \boldsymbol{\theta}(\mathbf{y})|^2\quad (4.143)$$

where $\boldsymbol{\theta}(\mathbf{y})$ is an estimate based on the observation \mathbf{y} .

4.6.7 BAYESIAN CLASSIFICATION OF FINITE STATE PROCESSES

In this section, the classification problem is formulated within the framework of a finite state random process. A finite state process is composed of a probabilistic chain of a number of different random processes. Finite state processes are used for modelling non-stationary signals such as speech, image, background acoustic noise, and impulsive noise as discussed in Chapter 5.

Consider a process with a set of M states denoted $S = \{s_1, s_2, \dots, s_M\}$, where each state has some distinct statistical property. In its simplest form, a state is just a single vector, and the finite state process is equivalent to a discrete-valued random process with M outcomes. In this case the Bayesian state estimation is identical to the Bayesian classification of a signal into one of M discrete-valued vectors. More generally, a state generates continuous-valued or discrete-valued vectors from a pdf or a pmf, associated with the state. Figure 4.19 illustrates an M -state process, where the output of the i th state is expressed as

$$\mathbf{x}(m) = h_i[\boldsymbol{\theta}_i, \mathbf{e}(m)], \quad i = 1, \dots, M \tag{4.144}$$

where in each state the signal $\mathbf{x}(m)$ is modelled as the output of a state-dependent function $h_i(\cdot)$ with parameter $\boldsymbol{\theta}_i$, input $\mathbf{e}(m)$ and an input pdf $f_{Ei}[\mathbf{e}(m)]$. Each state may be a model of a segment of speech or image. The prior probability of each state is given by

$$P_s(s_i) = \mathcal{E}[N(s_i)] / \mathcal{E} \left[\sum_{j=1}^M N(s_j) \right] \tag{4.145}$$

where $\mathcal{E}[N(s_i)]$ is the expected number of observations from state s_i . The pdf of the output of a finite state process is a weighted combination of the pdf of each state and is given by

$$f_X[\mathbf{x}(m)] = \sum_{i=1}^M P_s(s_i) f_{X|S}(\mathbf{x}|s_i) \tag{4.146}$$

In Figure 4.19, the noisy observation $\mathbf{y}(m)$ is the sum of the process output $\mathbf{x}(m)$ and an additive noise $\mathbf{n}(m)$. From Bayes' rule, the posterior probability of the state s_i given the

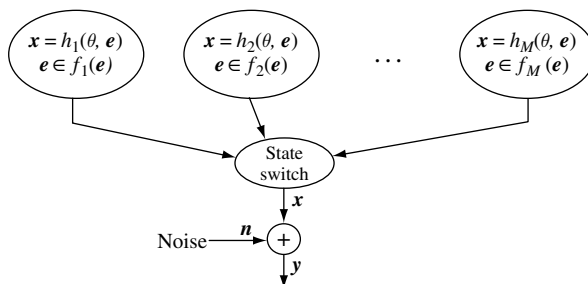


Figure 4.19 Illustration of a random process generated by a finite state system.

observation $\mathbf{y}(m)$ can be expressed as

$$P_{S|Y}[s_i|\mathbf{y}(m)] = \frac{f_{Y|S}[\mathbf{y}(m)|s_i]P_S(s_i)}{\sum_{j=1}^M f_{Y|S}[\mathbf{y}(m)|s_j]P_S(s_j)} \quad (4.147)$$

In MAP classification, the state with the maximum posterior probability is selected as

$$s_{\text{MAP}}[\mathbf{y}(m)] = \arg \max_{s_i} P_{S|Y}[s_i|\mathbf{y}(m)] \quad (4.148)$$

The Bayesian state classifier assigns a misclassification cost function $C(s_i|s_j)$ to the action of selecting the state s_i when the true state is s_j . The risk function for the Bayesian classification is given by

$$\mathcal{R}[s_i|\mathbf{y}(m)] = \sum_{j=1}^M C(s_i|s_j)P_{S|Y}[s_j|\mathbf{y}(m)] \quad (4.149)$$

4.6.8 BAYESIAN ESTIMATION OF THE MOST LIKELY STATE SEQUENCE

Consider the estimation of the most likely state sequence, $\mathbf{s} = [s_{i_0}, s_{i_1}, \dots, s_{i_{T-1}}]$, of a finite state process, given a sequence of T observation vectors, $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{T-1}]$. A state sequence, \mathbf{s} , of length T , is itself a random integer-valued vector process with N^T possible values. From the Bayes rule, the posterior pmf of a state sequence \mathbf{s} , given an observation sequence \mathbf{Y} , can be expressed as

$$P_{S|Y}(s_{i_0}, \dots, s_{i_{T-1}}|\mathbf{y}_0, \dots, \mathbf{y}_{T-1}) = \frac{f_{Y|S}(\mathbf{y}_0, \dots, \mathbf{y}_{T-1} | s_{i_0}, \dots, s_{i_{T-1}})P_S(s_{i_0}, \dots, s_{i_{T-1}})}{f_Y(\mathbf{y}_0, \dots, \mathbf{y}_{T-1})} \quad (4.150)$$

where $P_S(\mathbf{s})$ is the pmf of the state sequence \mathbf{s} , and for a given observation sequence the denominator $f_Y(\mathbf{y}_0, \dots, \mathbf{y}_{T-1})$ is a constant. The Bayesian risk in selecting a state sequence s_i is expressed as

$$\mathcal{R}(s_i|\mathbf{y}) = \sum_{j=1}^{N^T} C(s_i|s_j)P_{S|Y}(s_j|\mathbf{y}) \quad (4.151)$$

For a statistically independent process, the state of the process at any time is independent of the previous states, and hence the conditional probability of a state sequence can be written as

$$P_{S|Y}(s_{i_0}, \dots, s_{i_{T-1}}|\mathbf{y}_0, \dots, \mathbf{y}_{T-1}) = \prod_{k=0}^{T-1} f_{Y|S}(\mathbf{y}_k | s_{i_k})P_S(s_{i_k}) \quad (4.152)$$

where s_{i_k} denotes state s_i at time instant k . A particular case of a finite state process is the Markov chain, Figure 4.20, where the state transition is governed by a Markovian process

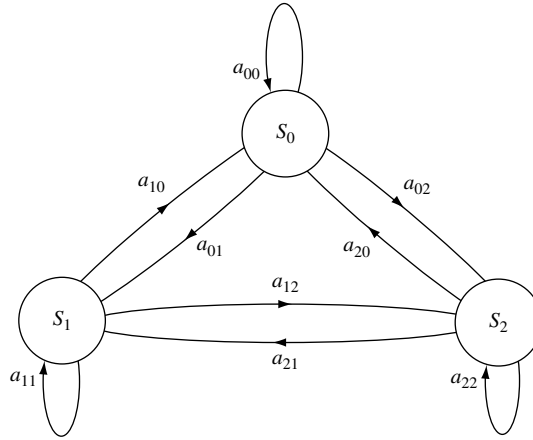


Figure 4.20 A three-state Markov process.

such that the probability of the state i at time m depends on the state of the process at time $m - 1$. The conditional pmf of a Markov state sequence can be expressed as

$$P_{S|Y}(s_{i_0}, \dots, s_{i_{T-1}} | y_0, \dots, y_{T-1}) = \prod_{k=0}^{T-1} a_{i_{k-1}i_k} f_{S|Y}(s_{i_k} | y_k) \quad (4.153)$$

where $a_{i_{k-1}i_k}$ is the probability that the process moves from state $S_{i_{k-1}}$ to state s_{i_k} . Finite state random processes and computationally efficient methods of state sequence estimation are described in detail in Chapter 5.

4.7 MODELLING THE SPACE OF A RANDOM PROCESS

In this section, we consider the training of statistical models for a database of P -dimensional vectors of a random process. The vectors in the database can be visualised as forming a number of clusters in a P -dimensional space. The statistical modelling method consists of two steps: (a) the partitioning of the database into a number of regions, or clusters, and (b) the estimation of the parameters of a statistical model for each cluster. A simple method for modelling the space of a random signal is to use a set of prototype vectors that represent the centroids of the signal space. This method effectively quantises the space of a random process into a relatively small number of typical vectors, and is known as *vector quantisation* (VQ). In the following, we first consider a VQ model of a random process, and then extend this model to a pdf model, based on a mixture of Gaussian densities.

4.7.1 VECTOR QUANTISATION OF A RANDOM PROCESS

Vector quantisations are used in signal compression and pattern recognition, such as in the coding or recognition of speech, music or image signals.

In vector quantisation, the space of the training data, from a random vector process \mathbf{X} , is partitioned into K clusters or regions $[X_1, X_2, \dots, X_K]$ and each cluster, X_i , is represented

by a cluster centroid, \mathbf{c}_i . The set of centroid vectors $[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K]$ form a VQ codebook model the process \mathbf{X} .

The VQ codebook can then be used to classify an unlabelled vector, \mathbf{x} , with the nearest centroid. The codebook is searched to find the centroid vector with the minimum distance from \mathbf{x} , then \mathbf{x} is labelled with the index of the minimum distance centroid as

$$\text{Label}(\mathbf{x}) = \arg \min_i d(\mathbf{x}, \mathbf{c}_i) \quad (4.154)$$

where $d(\mathbf{x}, \mathbf{c}_i)$ is a measure of distance between the vectors \mathbf{x} and \mathbf{c}_i . The most commonly used distance measure is the mean squared distance.

4.7.2 VECTOR QUANTISATION USING GAUSSIAN MODELS

In vector quantisation, instead of using only the centre of each cluster, a Gaussian pdf model of each cluster comprising of the centre of the cluster, its covariance matrix and its probability may be used. In this way, the space of the training data, from a random vector process \mathbf{X} , is partitioned into K clusters or regions $[\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K]$ and each cluster, \mathbf{X}_i , is represented by a cluster centroid, \mathbf{c}_i , the cluster covariance matrix, $\boldsymbol{\Sigma}_i$, and the cluster probability, p_i , as $[\mathcal{N}(\mathbf{c}_i, \boldsymbol{\Sigma}_i), p_i]$. The set of VQ pdfs $\{[\mathcal{N}(\mathbf{c}_1, \boldsymbol{\Sigma}_1), p_1], [\mathcal{N}(\mathbf{c}_2, \boldsymbol{\Sigma}_2), p_2], \dots, [\mathcal{N}(\mathbf{c}_K, \boldsymbol{\Sigma}_K), p_K]\}$ forms a VQ codebook to model the process \mathbf{X} . The VQ codebook can then be used to classify an unlabelled vector \mathbf{x} with the nearest pdf. The codebook is searched to find the VQ pdf with the maximum probability of membership for \mathbf{x} , then \mathbf{x} is labelled with the index of the pdf as

$$\text{Label}(\mathbf{x}) = \arg \max_i p_i \mathcal{N}(\mathbf{x}, \mathbf{c}_i, \boldsymbol{\Sigma}_i) \quad (4.155)$$

where the weighted Gaussian pdf distance $p_i \mathcal{N}(\mathbf{x}, \mathbf{c}_i, \boldsymbol{\Sigma}_i)$, is a measure of membership of the input vector, \mathbf{x} , and the VQ class, i .

4.7.3 DESIGN OF A VECTOR QUANTISER: K-MEANS CLUSTERING

The K -means algorithm, illustrated in Figure 4.21, is an iterative method for the design of a VQ codebook. Each iteration consists of two basic steps: (a) partition the training signal space into K regions or clusters and (b) compute the centroid of each region. The steps in K -Means method are as follows:

- Step 1: initialisation* – use a suitable method to choose a set of K initial centroids $[\mathbf{c}_i]$.
For $m = 1, 2, \dots$
- Step 2: classification* – classify the training vectors $\{\mathbf{x}\}$ into K clusters $\{[\mathbf{x}_1], [\mathbf{x}_2], \dots, [\mathbf{x}_K]\}$ using the so-called ‘nearest-neighbour rule’ [Equation (4.154)].
- Step 3: centroid computation* – use the vectors $[\mathbf{x}_i]$ associated with the i th cluster to compute an updated cluster centroid, \mathbf{c}_i , and calculate the cluster distortion defined as

$$D_i(m) = \frac{1}{N_i} \sum_{j=1}^{N_i} d[\mathbf{x}_i(j), \mathbf{c}_i(m)] \quad (4.156)$$

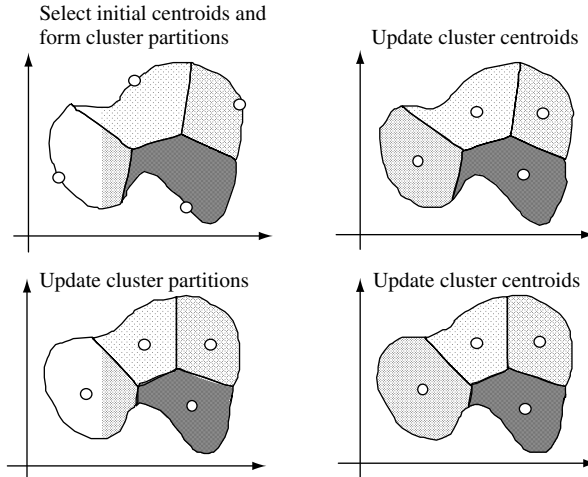


Figure 4.21 Illustration of the K -means clustering method.

where it is assumed that a set of N_i vectors $[x_i(j)j = 0, \dots, N_i]$ are associated with cluster i . The total distortion is given by

$$D(m) = \sum_{i=1}^K D_i(m) \quad (4.157)$$

Step 4: convergence test:

if
 $D(m-1) - D(m) \geq \text{threshold}$, stop,
 else
 goto Step 2.

A vector quantiser models the regions, or the clusters, of the signal space with a set of cluster centroids. A more complete description of the signal space can be achieved by modelling each cluster with a Gaussian density as described in Chapter 5.

4.8 SUMMARY

This chapter began with an introduction to the basic concepts in estimation theory; such as the signal space and the parameter space, the prior and posterior spaces, and the statistical measures that are used to quantify the performance of an estimator. The Bayesian inference method, with its ability to include as much information as is available, provides a general framework for statistical signal processing problems. The minimum mean square error, the maximum-likelihood, the maximum *a posteriori* and the minimum absolute value of error methods were derived from the Bayesian formulation. Further examples of the

applications of Bayesian type models in this book include the hidden Markov models for nonstationary processes studied in Chapter 5, and blind equalisation of distorted signals studied in Chapter 15.

We considered a number of examples of the estimation of a signal observed in noise, and derived the expressions for the effects of using prior pdfs on the mean and the variance of the estimates. The choice of the prior pdf is an important consideration in Bayesian estimation. Many processes, for example speech or the response of a telecommunications channel, are not uniformly distributed in space, but are constrained to a particular region of signal or parameter space. The use of a prior pdf can guide the estimator to focus on the posterior space that is the subspace consistent with both the likelihood and the prior pdfs. The choice of the prior, depending on how well it fits the process, can have a significant influence on the solutions.

The iterative EM method, studied in Section 4.3, provides a practical framework for solving many statistical signal processing problems, such as the modelling of a signal space with a mixture Gaussian densities, and the training of hidden Markov models in Chapter 5. In Section 4.4 the Cramer–Rao lower bound on the variance of an estimator was derived, and it was shown that the use of a prior pdf can reduce the minimum estimator variance.

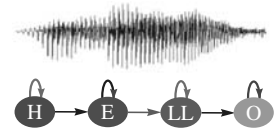
Finally we considered the modelling of a data space with a mixture Gaussian process, and used the EM method to derive a solution for the parameters of the mixture Gaussian model.

BIBLIOGRAPHY

- Abramson N. (1963) *Information Theory and Coding*. McGraw-Hill, New York.
- Andergerg M.R. (1973) *Cluster Analysis for Applications*. Academic Press, New York.
- Baum L.E., Petrie T., Soules G. and Weiss N. (1970) A maximisation technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.* **41**: 164–171.
- Bayes T. (1763) An essay towards solving a problem in the doctrine of changes. *Phil. Trans. R. Soc. Lond.* **53**: 370–418 (reprinted in 1958 in *Biometrika* **45**: 293–315).
- Chou P. Lookabaugh T. and Gray R. (1989) Entropy-constrained vector quantisation. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-37**: 31–42.
- Bezdek J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.
- Cramer H. (1974) *Mathematical Methods of Statistics*. Princeton University Press, Princeton, NJ.
- Dempster A.P., Laird N.M. and Rubin D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. Ser. B* **39**: 1–38.
- Deutsch R. (1965) *Estimation Theory*. Prentice-Hall, Englewood Cliffs, NJ.
- Duda R.O. and Hart R.E. (1973) *Pattern Classification*. Wiley, New York.
- Feder M. and Weinstein E. (1988) Parameter estimation of superimposed signals using the EM algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-36(4)**: 477–489.
- Fisher R.A. (1922) On the mathematical foundations of the theoretical statistics. *Phil. Trans. R. Soc. Lond.* **222**: 309–368.
- Jersho A. (1982) On the structure of vector quantisers. *IEEE Trans. Inform. Theory* **IT-28**: 157–166.
- Gray R.M. (1984) Vector quantisation. *IEEE ASSP Mag.*, 4–29.
- Gray R.M. and Karnin E.D. (1982), Multiple local optima in vector quantisers. *IEEE Trans. Inform. Theory* **IT-28**: 256–261.
- Jeffrey H. (1961) *Scientific Inference*, 3rd edn. Cambridge University Press, Cambridge.
- Larson H.J. and Bruno O.S. (1979) *Probabilistic Models in Engineering Sciences. I and II*. Wiley, New York.
- Linde Y., Buzo A. and Gray R.M. (1980) An algorithm for vector quantiser design. *IEEE Trans. Commun.* **COM-28**: 84–95.
- Makhoul J., Roucos S. and Gish H. (1985) Vector quantisation in speech coding. *Proc. IEEE* **73**: 1551–1588.

- Mohanty N. (1986) *Random Signals, Estimation and Identification*. Van Nostrand, New York.
- Rao C.R. (1945) Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.* **37**: 81–91.
- Render R.A. and Walker H.F. (1984) Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **26**: 195–239.
- Scharf L.L. (1991) *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison Wesley, Reading, MA.

5



Hidden Markov Models

5.1 Statistical Models for Nonstationary Processes	138
5.2 Hidden Markov Models	139
5.3 Training Hidden Markov Models	145
5.4 Decoding of Signals Using Hidden Markov Models	152
5.5 HMMs in DNA and Protein Sequence Modelling	155
5.6 HMMs for Modelling Speech and Noise	156
5.7 Summary	162
Bibliography	163

Hidden Markov models are used for the statistical modelling of nonstationary signal processes such as speech signals, image sequences and time-varying noise. The Markov process, developed by Andri Markov, is a process whose state or value at any time, t , depends on its previous state and values at time $t - 1$, and is independent of the history of the process before $t - 1$. An HMM is a double-layered process with a ‘hidden’ Markov layer controlling the state of an observable layer.

An HMM models the time variations (and/or the space variations) of the statistics of a random process with a Markovian chain of state-dependent stationary subprocesses. An HMM is essentially a Bayesian finite state process, with a Markovian prior for modelling the transitions between the states, and a set of state probability density functions for modelling the random variations of the signal process within each state. This chapter begins with a brief introduction to continuous and finite state nonstationary models, before concentrating on the theory and applications of hidden Markov models. We study the various HMM structures, the Baum–Welch method for the maximum-likelihood training of the parameters of an HMM, and the use of HMMs and the Viterbi decoding algorithm for the classification and decoding of an unlabelled observation signal sequence. Finally, applications of the HMMs for the enhancement of noisy signals are considered.

5.1 STATISTICAL MODELS FOR NONSTATIONARY PROCESSES

A nonstationary process can be defined as one whose statistical parameters vary over time. Most ‘naturally generated’ signals, such as audio signals, video signals, biomedical signals and seismic signals, are nonstationary, in that the parameters of the systems that generate the signals, and the environments in which the signals propagate, change with time and/or space.

A nonstationary process can be modelled as a double-layered stochastic process, with a hidden process that controls the time variations of the statistics of an observable process, as illustrated in Figure 5.1. In general, nonstationary processes can be classified into one of two broad categories:

- (1) *continuously variable state* processes; and
- (2) *finite state* processes.

A continuously variable state process is defined as one whose underlying statistics vary continuously with time. Examples of this class of random processes are some audio signals such as speech, whose power and spectral composition, within a phonetic segment, vary continuously with time. A finite state process is one whose statistical characteristics can *switch* between a finite number of stationary or nonstationary states. For example, impulsive noise is a binary-state process and across different phonetic segments speech is a finite state process. Continuously variable processes can be approximated by an appropriate finite state process.

Figure 5.2(a) illustrates a nonstationary first-order AR process. This process is modelled as the combination of a *hidden* stationary AR model of the signal parameters and an observable time-varying AR model of the signal. The hidden model controls the time variations of the parameters of the nonstationary AR model. For this model, the observation signal equation and the hidden parameter state equation can be expressed as

$$x(m) = a(m)x(m-1) + e(m) \quad \text{observation equation} \quad (5.1)$$

$$a(m) = \beta a(m-1) + \varepsilon(m) \quad \text{hidden state equation} \quad (5.2)$$

where $a(m)$ is the time-varying coefficient of the observable AR process and β is the coefficient of the hidden state-control process.

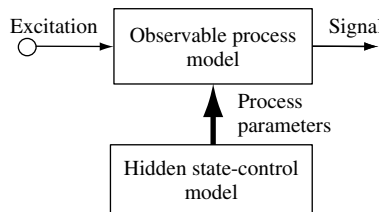


Figure 5.1 Illustration of a two-layered model of a nonstationary process.

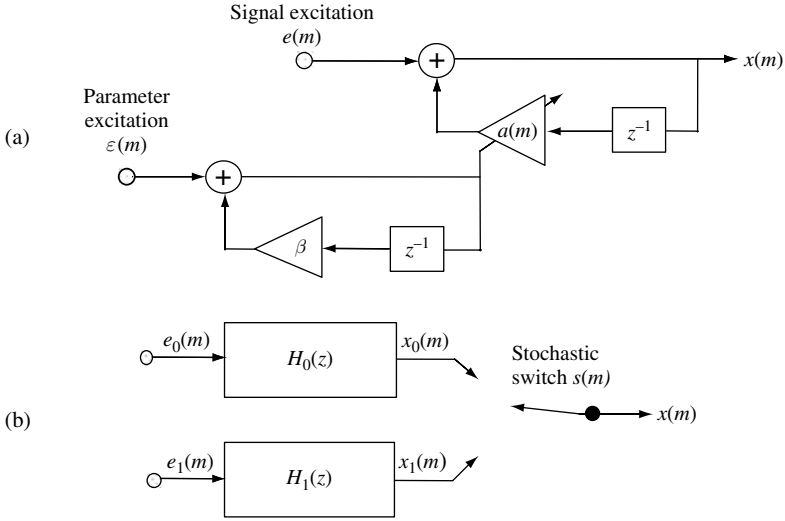


Figure 5.2 (a) A continuously variable state AR process. (b) A binary-state AR process.

A simple example of a finite state nonstationary model is the binary-state AR process illustrated in Figure 5.2(b), where at each time instant a random switch selects one of the two AR models for connection to the output terminal. For this model, the output signal, $x(m)$, can be expressed as

$$x(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \tag{5.3}$$

where the binary switch, $s(m)$, selects the state of the process at time m , and $\bar{s}(m)$ denotes the Boolean complement of $s(m)$.

5.2 HIDDEN MARKOV MODELS

5.2.1 COMPARISON OF MARKOV AND HIDDEN MARKOV MODELS

A Markov process is defined as stochastic process whose probability of being in a given state at time m depends on the pervious state of the system at time $m - 1$ and is independent of the states of the process before $m - 1$.

Consider a simple example of the two-state Markov process illustrated in Figure 5.3(a), which shows two containers (states): in state 1 the process outputs black balls and in state 2 the process outputs white balls. Assume that at successive time intervals a random selection process selects one of the two containers to release a ball. The state selection process is probabilistic such that in state 1 the probability of staying in state 1 at time m is $p(s_{1,m}|s_{1,m-1}) = 0.8$, whereas the probability of moving to state 2 is $p(s_{2,m}|s_{1,m-1}) = 0.2$, where $s_{i,m}$ denotes state i at time m . In state 2 the probability of staying in state 2 is $p(s_{2,m}|s_{2,m-1}) = 0.6$, whereas the probability moving from state 2 to state 1 is $p(s_{1,m}|s_{2,m-1}) = 0.4$. Note that the Markov process output sequence is the same as the state sequence.

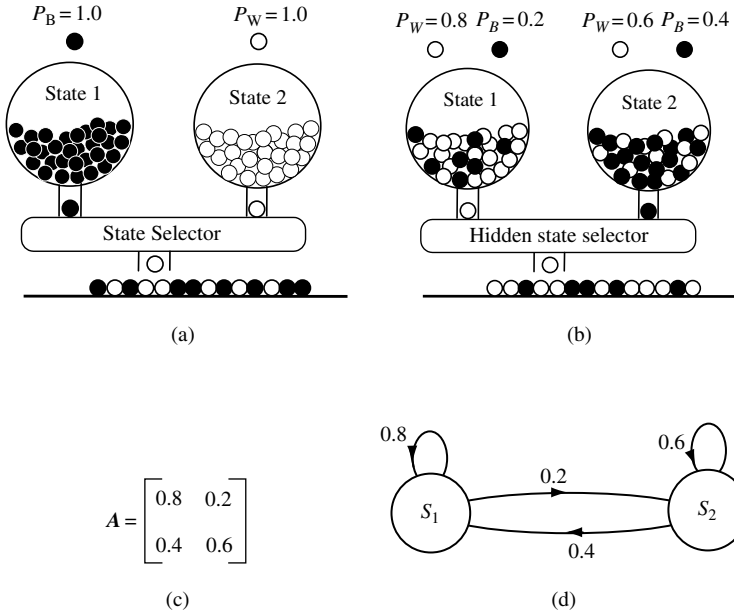


Figure 5.3 (a) A Markov model: each state is identified by the output. (b) A hidden Markov model: states are ‘hidden’ as both states can produce the same output with different probability. (c) The assumed transition matrix for (a) and (b). (d) A state diagram for Markov model and HMM.

A HMM is a double-layered finite state process, with a hidden Markovian process that controls the selection of the states of an observable process. As a simple illustration of a binary-state Markovian process, consider Figure 5.3(b), which shows two containers of different mixtures of black and white balls. The probability of the black and the white balls in each container, denoted P_B and P_W , is as shown in Figure 5.3(b). Assume that at successive time intervals a hidden selection process selects one of the two containers to release a ball. The balls released are replaced so that the mixture density of black and white balls in each container remains unaffected. Each container can be considered as an underlying state of the output process. Now, as an example assume that the hidden container-selection process is governed by the following rule: at any time, if the output from the currently selected container is a white ball then the same container is selected to output the next ball, otherwise the other container is selected. This is an example of a Markovian process because the next state of the process depends on the current state, as shown in the binary state model of Figure 5.3(d). Note that in this example the observable outcome does not unambiguously indicate the underlying hidden state, because both states are capable of releasing black and white balls.

In general, a hidden Markov model has N states, with each state trained to model a distinct segment of a signal process. A hidden Markov model can be used to model a time-varying random process as a probabilistic Markovian chain of N stationary, or quasistationary, elementary subprocesses. A general form of a three-state HMM is shown in Figure 5.4. This structure is known as an *ergodic* HMM. In the context of an HMM, the

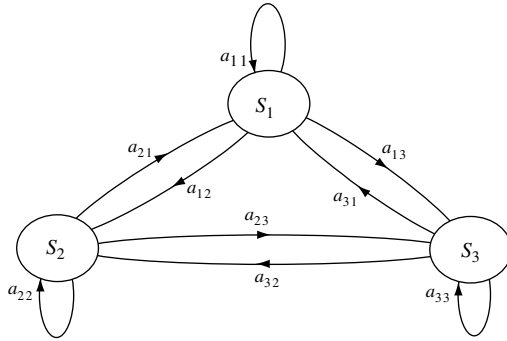


Figure 5.4 A three-state ergodic HMM structure.

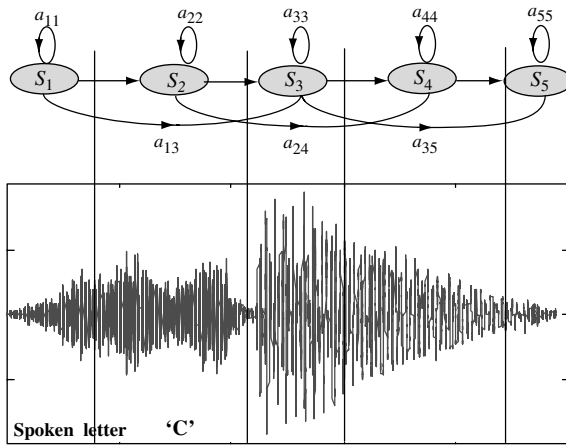


Figure 5.5 A five-state left-right HMM speech model.

term ‘ergodic’ implies that there are no structural constraints for connecting any state to any other state.

A more constrained form of an HMM is the left-right model of Figure 5.5, so-called because the allowed state transitions are those from a left state to a right state and the self-loop transitions. The left-right constraint is useful for the characterisation of temporal or sequential structures of stochastic signals such as speech and musical signals, because time may be visualised as having a direction from left to right.

5.2.2 A PHYSICAL INTERPRETATION: HMMs OF SPEECH

For a physical interpretation of the use of HMMs in modelling a signal process, consider the illustration of Figure 5.5 which shows a left-right HMM of a spoken letter ‘C’, phonetically transcribed ‘s-iy’, together with a plot of the speech signal waveform for ‘C’. In general,

there are two main types of variation in speech and other stochastic signals: variations in the spectral composition, and variations in the time-scale or the articulation rate. In a hidden Markov model, these variations are modelled by the state observation and the state transition probabilities.

A useful way of interpreting and using HMMs is to consider each state of an HMM as a model of a segment of a stochastic process. For example, in Figure 5.5, state S_1 models the first segment of the spoken letter 'C', state S_2 models the second segment, and so on. Each state must have a mechanism to accommodate the random variations in different realisations of the segments that it models. The state transition probabilities provide a mechanism for connection of various states, and for modelling the variations in the duration and time-scales of the signals in each state. For example, if a segment of a speech utterance is elongated, owing, say, to slow articulation, then this can be accommodated by more self-loop transitions into the state that models the segment. Conversely, if a segment of a word is omitted, owing, say, to fast speaking, then the skip-next-state connection accommodates that situation. The state observation pdfs model the space of the probability distributions of the spectral composition of the signal segments associated with each state.

5.2.3 HIDDEN MARKOV MODEL AS A BAYESIAN MODEL

A hidden Markov model, \mathcal{M} , is a Bayesian structure with a Markovian state transition probability and a state observation likelihood that can be either a discrete pmf or a continuous pdf.

The *posterior* probability of a state sequence, s , of a model, \mathcal{M} , given a sequence of observation vectors, $\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T-1)]$, can be expressed using Bayes's rule as the product of the *prior* probability of the state sequence, s , and the *likelihood* of the observation, \mathbf{X} , as

$$P_{S|\mathbf{X}, \mathcal{M}}(s|\mathbf{X}, \mathcal{M}) = \frac{1}{f_{\mathbf{X}}(\mathbf{X})} \underbrace{P_{S|\mathcal{M}}(s|\mathcal{M})}_{\text{State prior}} \underbrace{f_{\mathbf{X}|S, \mathcal{M}}(\mathbf{X}|s, \mathcal{M})}_{\text{Observation likelihood}} \quad (5.4)$$

where the observation sequence, \mathbf{X} , is modelled by a probability density function $P_{\mathbf{X}|S, \mathcal{M}}(\mathbf{X}|s, \mathcal{M})$.

The posterior probability that an observation signal sequence, \mathbf{X} , was generated by the model, \mathcal{M} , is summed over all likely state sequences, and may also be weighted by the model prior, $P_{\mathcal{M}}(\mathcal{M})$:

$$P_{\mathcal{M}|\mathbf{X}}(\mathcal{M}|\mathbf{X}) = \frac{1}{f_{\mathbf{X}}(\mathbf{X})} \underbrace{P_{\mathcal{M}}(\mathcal{M})}_{\text{Model Prior}} \sum_s \underbrace{P_{S|\mathcal{M}}(s|\mathcal{M})}_{\text{State Prior}} \underbrace{f_{\mathbf{X}|S, \mathcal{M}}(\mathbf{X}|s, \mathcal{M})}_{\text{Observation Likelihood}} \quad (5.5)$$

The Markovian state transition prior can be used to model the time variations and the sequential dependence of most nonstationary processes. However, for many applications, such as speech recognition, the state observation likelihood has far more influence on the posterior probability than the state transition prior.

5.2.4 PARAMETERS OF A HIDDEN MARKOV MODEL

A hidden Markov model has the following parameters:

- *Number of states* N – this is usually set to the total number of distinct, or elementary, stochastic events in a signal process. For example, in modelling a binary-state process such as impulsive noise, N is set to 2, and in phoneme-based speech modelling, N , the number of states for each phoneme, is set between 3 and 5.
- *State transition-probability matrix* $\mathbf{A} = \{a_{ij}, i, j = 1, \dots, N\}$ – this provides a Markovian connection network between the states, and models the variations in the duration of the signals associated with each state. For a left–right HMM (see Figure 5.5), $a_{ij} = 0$ for $i > j$, and hence the transition matrix \mathbf{A} is upper-triangular.
- *State observation vectors* $\{\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \dots, \boldsymbol{\mu}_{iM}, i = 1, \dots, N\}$ – for each state a set of M prototype vectors models the centroids of the signal space associated with that state.
- *State observation vector probability model* – this can be either a discrete model composed of M prototype vectors and their associated probability $\mathbf{P} = \{P_{ij}(\cdot); i = 1, \dots, N, j = 1, \dots, M\}$, or it may be a continuous (usually Gaussian) pdf model $\mathbf{F} = \{f_{ij}(\cdot); i = 1, \dots, N, j = 1, \dots, M\}$.
- *Initial state probability vector* $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_N]$.

5.2.5 STATE OBSERVATION PROBABILITY MODELS

Depending on whether a signal process is discrete-valued or continuous-valued, the state observation model for the process can be either a discrete-valued probability mass function (pmf), or a continuous-valued probability density function (pdf). The discrete models can also be used for the modelling of the space of a continuous-valued process quantised into a number of discrete points.

First, consider a discrete state observation density model. Assume that associated with the i th state of an HMM there are M discrete centroid vectors $[\boldsymbol{\mu}_{i1}, \dots, \boldsymbol{\mu}_{iM}]$ with a pmf $[P_{i1}, \dots, P_{iM}]$. These centroid vectors and their probabilities are normally obtained through clustering of a set of training signals associated with each state.

For the modelling of a continuous-valued process, the signal space associated with each state is partitioned into a number of clusters, as in Figure 5.6. If the signals within each cluster are modelled by a uniform distribution, then each cluster is described by the centroid vector and the cluster probability, and the state observation model consists of M cluster centroids and the associated pmf $\{\boldsymbol{\mu}_{ik}, P_{ik}; i = 1, \dots, N, k = 1, \dots, M\}$. In effect, this results in a discrete observation HMM for a continuous-valued process.

Figure 5.6(a) shows a partitioning, and quantisation, of a signal space into a number of centroids. Now if each cluster of the state observation space is modelled by a continuous pdf, such as a Gaussian pdf, then a continuous-density HMM results. The most widely used state observation pdf for an HMM is the Gaussian mixture density, defined as

$$f_{X|S}(\mathbf{x}|s=i) = \sum_{k=1}^M P_{ik} \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (5.6)$$

where $\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})$ is a Gaussian density with mean vector $\boldsymbol{\mu}_{ik}$ and covariance matrix $\boldsymbol{\Sigma}_{ik}$, and P_{ik} is a mixture weighting factor for the k th Gaussian pdf of the state i . Note that P_{ik} is

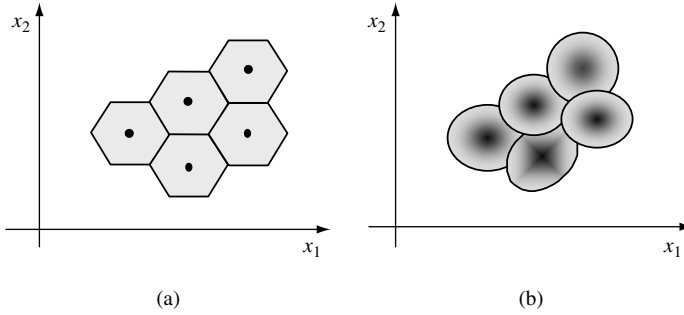


Figure 5.6 Modelling a random signal space using (a) a discrete-valued pmf and (b) a continuous-valued mixture Gaussian density.

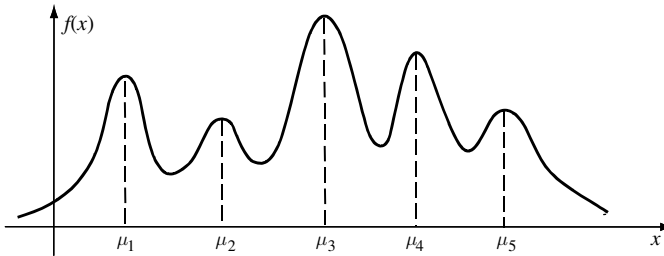


Figure 5.7 A mixture Gaussian probability density function.

the prior probability of the k th mode of the pdf mixture for the state i . Figure 5.6(b) shows the space of a Gaussian mixture model of an observation signal space. A five-mode Gaussian mixture pdf is shown in Figure 5.7.

5.2.6 STATE TRANSITION PROBABILITIES

The first-order Markovian property of an HMM entails that the transition probability to any state, $s(t)$, at time t depends only on the state of the process at time $t - 1$, $s(t - 1)$, and is independent of the previous states of the HMM. This can be expressed as

$$\begin{aligned} \text{Prob}[s(t) = j | s(t - 1) = i, s(t - 2) = k, \dots, s(t - N) = l] \\ = \text{Prob}[s(t) = j | s(t - 1) = i] = a_{ij} \end{aligned} \quad (5.7)$$

where $s(t)$ denotes the state of HMM at time t . The transition probabilities provide a probabilistic mechanism for connecting the states of an HMM, and for modelling the variations in the duration of the signals associated with each state. The probability of occupancy of a state i for d consecutive time units, $P_i(d)$, can be expressed in terms of the state self-loop transition probabilities a_{ii} as

$$P_i(d) = a_{ii}^{d-1} (1 - a_{ii}) \quad (5.8)$$

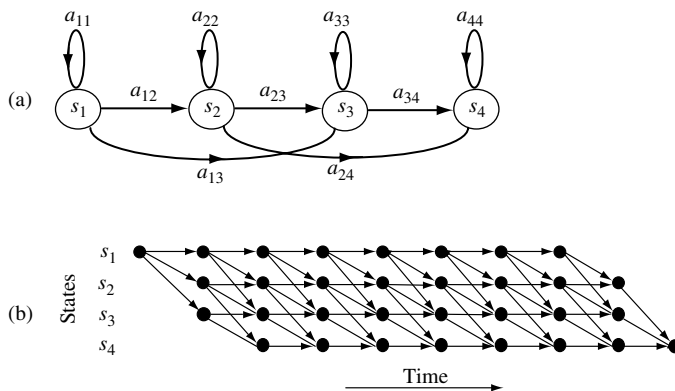


Figure 5.8 (a) A four-state left–right HMM, and (b) its state–time trellis diagram.

From Equation (5.8), using the geometric series conversion formula, the mean occupancy duration for each state of an HMM can be derived as

$$\text{Mean occupancy of state } i = \sum_{d=0}^{\infty} dP_i(d) = \frac{1}{1 - a_{ii}} \tag{5.9}$$

5.2.7 STATE–TIME TRELLIS DIAGRAM

A state–time trellis diagram shows the HMM states together with all the different paths that can be taken through various states as time unfolds. Figure 5.8 illustrates a four-state HMM and its state–time diagram. Since the number of states and the state parameters of an HMM are time-invariant, a state–time diagram is a repetitive and regular trellis structure. Note that in Figure 5.8, for a left–right HMM, the state–time trellis has to diverge from the first state and converge into the last state. In general, there are many different state sequences that start from the initial state and end in the final state. Each state sequence has a prior probability that can be obtained by multiplication of the state transition probabilities of the sequence. For example, the probability of the state sequence $s = [S_1, S_1, S_2, S_2, S_3, S_3, S_4]$ is $P(s) = \pi_1 a_{11} a_{12} a_{22} a_{23} a_{33} a_{34}$. Since each state has a different set of prototype observation vectors, different state sequences model different observation sequences. In general, over T time units, an N -state HMM can reproduce N^T different realisations of the random process of length T .

5.3 TRAINING HIDDEN MARKOV MODELS

The first step in training the parameters of an HMM is to collect a training database of a sufficiently large number of different examples of the random process to be modelled. Assume that the examples in a training database consist of L vector-valued sequences, $[X] = [X_k; k = 0, \dots, L - 1]$, with each sequence $X_k = [x(t); t = 0, \dots, T_k - 1]$ having a variable number of T_k vectors. The objective is to train the parameters of an HMM to model

the statistics of the signals in the training data set. In a probabilistic sense, the fitness of a model is measured by the posterior probability $P_{\mathcal{M}|\mathbf{X}}(\mathcal{M}|\mathbf{X})$ of the model \mathcal{M} given the training data \mathbf{X} . The training process aims to maximise the posterior probability of the model \mathcal{M} and the training data $[\mathbf{X}]$, expressed using Bayes's rule as

$$P_{\mathcal{M}|\mathbf{X}}(\mathcal{M}|\mathbf{X}) = \frac{1}{f_{\mathbf{X}}(\mathbf{X})} f_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M}) P_{\mathcal{M}}(\mathcal{M}) \quad (5.10)$$

where the denominator $f_{\mathbf{X}}(\mathbf{X})$ on the right-hand side of Equation (5.10) has only a normalising effect and $P_{\mathcal{M}}(\mathcal{M})$ is the prior probability of the model \mathcal{M} . For a given training data set $[\mathbf{X}]$ and a given model \mathcal{M} , maximising Equation (5.10) is equivalent to maximising the likelihood function $P_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M})$. The likelihood of an observation vector sequence, \mathbf{X} , given a model \mathcal{M} can be expressed as

$$P_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M}) = \sum_{\mathbf{s}} f_{\mathbf{X}|\mathbf{s},\mathcal{M}}(\mathbf{X}|\mathbf{s},\mathcal{M}) P_{\mathbf{s}|\mathcal{M}}(\mathbf{s}|\mathcal{M}) \quad (5.11)$$

where $f_{\mathbf{X}|\mathbf{s},\mathcal{M}}(\mathbf{X}(t)|s(t),\mathcal{M})$, the pdf of the signal sequence \mathbf{X} along the state sequence $\mathbf{s} = [s(0), s(1), \dots, s(T-1)]$ of the model \mathcal{M} , is given by

$$f_{\mathbf{X}|\mathbf{s},\mathcal{M}}(\mathbf{X}|\mathbf{s},\mathcal{M}) = f_{X|S}[\mathbf{x}(0)|s(0)] f_{X|S}[\mathbf{x}(1)|s(1)] \dots f_{X|S}[\mathbf{x}(T-1)|s(T-1)] \quad (5.12)$$

where $s(t)$, the state at time t , can be one of N states, and $f_{X|S}[\mathbf{X}(t)|s(t)]$, a shorthand for $f_{\mathbf{X}|\mathbf{s},\mathcal{M}}[\mathbf{X}(t)|s(t),\mathcal{M}]$, is the pdf of $\mathbf{x}(t)$ given the state $s(t)$ of the model \mathcal{M} . The Markovian probability of the state sequence, \mathbf{s} , is given by

$$P_{\mathbf{s}|\mathcal{M}}(\mathbf{s}|\mathcal{M}) = \pi_{s(0)} a_{s(0)s(1)} a_{s(1)s(2)} \dots a_{s(T-2)s(T-1)} \quad (5.13)$$

Substituting Equations (5.12) and (5.13) in Equation (5.11) yields

$$\begin{aligned} P_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M}) &= \sum_{\mathbf{s}} f_{\mathbf{X}|\mathbf{s},\mathcal{M}}(\mathbf{X}|\mathbf{s},\mathcal{M}) P_{\mathbf{s}|\mathcal{M}}(\mathbf{s}|\mathcal{M}) \\ &= \sum_{\mathbf{s}} \pi_{s(0)} f_{X|S}[\mathbf{x}(0)|s(0)] a_{s(0)s(1)} f_{X|S}[\mathbf{x}(1)|s(1)] \dots a_{s(T-2)s(T-1)} f_{X|S} \\ &\quad \times [\mathbf{x}(T-1)|s(T-1)] \end{aligned} \quad (5.14)$$

where the summation is taken over all state sequences \mathbf{s} . In the training process, the transition probabilities and the parameters of the observation pdfs are estimated to maximise the model likelihood of Equation (5.14). Direct maximisation of Equation (5.14) with respect to the model parameters is a nontrivial task. Furthermore, for an observation sequence of length T vectors, the computational load of Equation (5.14) is $O(N^T)$. This is an impractically large load, even for such modest values as $N = 6$ and $T = 30$. However, the repetitive structure of the trellis state–time diagram of an HMM implies that there is a large amount of repeated, redundant, computation in Equation (5.14) that can be avoided in an efficient implementation. In the next section we consider the forward–backward method of model likelihood calculation, and then proceed to describe an iterative maximum-likelihood model optimisation method.

5.3.1 FORWARD–BACKWARD PROBABILITY COMPUTATION

An efficient recursive algorithm for the computation of the likelihood function $f_{X|\mathcal{M}}(\mathbf{X}|\mathcal{M})$ is the forward–backward algorithm. The forward–backward computation method exploits the highly regular and repetitive structure of the state–time trellis diagram of Figure 5.8.

In this method, a forward probability variable $\alpha_t(i)$ is defined as the joint probability of the partial observation sequence $\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(t)]$ and the state i at time t , of the model \mathcal{M} :

$$\alpha_t(i) = f_{X,S|\mathcal{M}}[\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(t), s(t) = i|\mathcal{M}] \tag{5.15}$$

The forward probability variable, $\alpha_t(i)$, of Equation (5.15) can be expressed in a recursive form in terms of the forward probabilities at time $t - 1$, $\alpha_{t-1}(i)$:

$$\begin{aligned} \alpha_t(i) &= f_{X,S|\mathcal{M}}[\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(t), s(t) = i|\mathcal{M}] \\ &= \left\{ \sum_{j=1}^N f_{X,S|\mathcal{M}}[\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(t-1), s(t-1) = j|\mathcal{M}] a_{ji} \right\} f_{X|S,\mathcal{M}}[\mathbf{x}(t)|s(t) = i, \mathcal{M}] \\ &= \sum_{j=1}^N [\alpha_{t-1}(j) a_{ji}] f_{X|S,\mathcal{M}}[\mathbf{x}(t)|s(t) = i, \mathcal{M}] \end{aligned} \tag{5.16}$$

Figure 5.9 illustrates a network for computation of the forward probabilities for the four-state left–right HMM of Figure 5.8. The likelihood of an observation sequence, $\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T - 1)]$, given a model \mathcal{M} can be expressed in terms of the forward probabilities as

$$\begin{aligned} f_{X|\mathcal{M}}[\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T - 1)|\mathcal{M}] &= \sum_{i=1}^N f_{X,S|\mathcal{M}}[\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T - 1), s(T - 1) = i|\mathcal{M}] \\ &= \sum_{i=1}^N \alpha_{T-1}(i) \end{aligned} \tag{5.17}$$

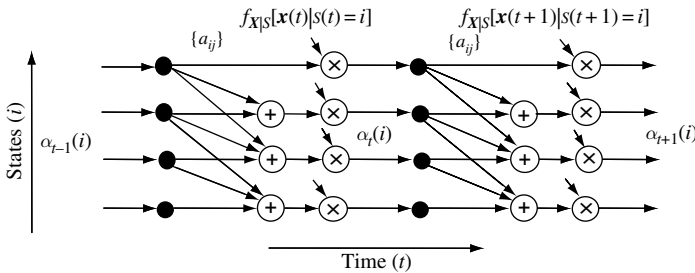


Figure 5.9 A network for computation of forward probabilities for a left–right HMM.

Similar to the definition of the forward probability concept, a backward probability is defined as the probability of the state i at time t followed by the partial observation sequence $[\mathbf{x}(t+1), \mathbf{x}(t+2), \dots, \mathbf{x}(T-1)]$ as

$$\begin{aligned}
 \beta_t(i) &= f_{\mathbf{X}, S | \mathcal{M}} [s(t) = i, \mathbf{x}(t+1), \mathbf{x}(t+2), \dots, \mathbf{x}(T-1) | \mathcal{M}] \\
 &= \sum_{j=1}^N a_{ij} f_{\mathbf{X}, S | \mathcal{M}} [s(t+1) = j, \mathbf{x}(t+2), \mathbf{x}(t+3), \dots, \mathbf{x}(T-1)] \\
 &\quad \times f_{\mathbf{X} | S} [\mathbf{x}(t+1) | s(t+1) = j | \mathcal{M}] \\
 &= \sum_{j=1}^N a_{ij} \beta_{t+1}(j) f_{\mathbf{X} | S, \mathcal{M}} [\mathbf{x}(t+1) | s(t+1) = j | \mathcal{M}]
 \end{aligned} \tag{5.18}$$

In the next section, forward and backward probabilities are used to develop a method for the training of HMM parameters.

5.3.2 BAUM–WELCH MODEL RE-ESTIMATION

The HMM training problem is the estimation of the model parameters $\mathcal{M} = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{F})$ for a given data set, \mathbf{X} . These parameters are the initial state probabilities $\boldsymbol{\pi}$, the state transition probability matrix \mathbf{A} and the continuous (or discrete) density state observation pdfs. The HMM parameters are estimated from a set of training examples $\{\mathbf{X} = [\mathbf{x}(0), \dots, \mathbf{x}(T-1)]\}$, with the objective of maximising $f_{\mathbf{X} | \mathcal{M}}(\mathbf{X} | \mathcal{M})$, the likelihood of the model and the training data. The Baum–Welch method of training HMMs is an iterative likelihood maximisation method based on the forward–backward probabilities defined in the preceding section. The Baum–Welch method is an instance of the EM algorithm described in Chapter 4. For an HMM \mathcal{M} , the posterior probability of a transition at time t from state i to state j of the model \mathcal{M} , given an observation sequence \mathbf{X} , can be expressed as

$$\begin{aligned}
 \gamma_t(i, j) &= P_{S | \mathbf{X}, \mathcal{M}} [s(t) = i, s(t+1) = j | \mathbf{X}, \mathcal{M}] \\
 &= \frac{f_{S, \mathbf{X} | \mathcal{M}} [s(t) = i, s(t+1) = j, \mathbf{X} | \mathcal{M}]}{f_{\mathbf{X} | \mathcal{M}} (\mathbf{X} | \mathcal{M})} \\
 &= \frac{\alpha_t(i) a_{ij} f_{\mathbf{X} | S, \mathcal{M}} [\mathbf{x}(t+1) | s(t+1) = j, \mathcal{M}] \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_{t-1}(i)}
 \end{aligned} \tag{5.19}$$

where $f_{S, \mathbf{X} | \mathcal{M}} [s(t) = i, s(t+1) = j, \mathbf{X} | \mathcal{M}]$ is the joint pdf of the states $s(t)$ and $s(t+1)$ and the observation sequence \mathbf{X} , and $f_{\mathbf{X} | S} [\mathbf{x}(t+1) | s(t+1) = i]$ is the state observation pdf for the state i . Note that, for a discrete observation density HMM, the state observation pdf in Equation (5.19) is replaced by the discrete state observation pmf, $P_{\mathbf{X} | S} [\mathbf{x}(t+1) | s(t+1) = i]$.

The posterior probability of state i at time t given the model, \mathcal{M} , and the observation, \mathbf{X} , is

$$\begin{aligned} \gamma_t(i) &= P_{S|\mathbf{X},\mathcal{M}}[s(t) = i|\mathbf{X}, \mathcal{M}] \\ &= \frac{f_{S,\mathbf{X}|\mathcal{M}}[s(t) = i, \mathbf{X}|\mathcal{M}]}{f_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M})} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_{T-1}(j)} \end{aligned} \tag{5.20}$$

Now the state transition probability a_{ij} can be interpreted as

$$a_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \tag{5.21}$$

From Equations (5.19)–(5.21), the state transition probability can be re-estimated as the ratio

$$\bar{a}_{ij} = \frac{\sum_{t=0}^{T-2} \gamma_t(i, j)}{\sum_{t=0}^{T-2} \gamma_t(i)} \tag{5.22}$$

Note that, for an observation sequence $[\mathbf{x}(0), \dots, \mathbf{x}(T-1)]$ of length T , the last transition occurs at time $T-2$, as indicated in the upper limits of the summations in Equation (5.22). The initial-state probabilities are estimated as

$$\bar{\pi}_i = \gamma_0(i) \tag{5.23}$$

5.3.3 TRAINING HMMs WITH DISCRETE DENSITY OBSERVATION MODELS

In a discrete density HMM, the observation signal space for each state is modelled by a set of discrete symbols or vectors. Assume that a set of M vectors $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \dots, \boldsymbol{\mu}_{iM}]$ model the space of the signal associated with the i th state. These vectors may be obtained from a clustering process as the centroids of the clusters of the training signals associated with each state. The objective in training discrete density HMMs is to compute the state transition probabilities and the state observation probabilities. The forward–backward equations for discrete density HMMs are the same as those for continuous density HMMs, derived in the previous sections, with the difference that the probability density functions such as $f_{\mathbf{X}|S}[\mathbf{x}(t)|s(t) = i]$ are substituted by probability mass functions $P_{\mathbf{X}|S}[\mathbf{x}(t)|s(t) = i]$ defined as

$$P_{\mathbf{X}|S}[\mathbf{x}(t)|s(t) = i] = P_{\mathbf{X}|S}\{\mathcal{Q}[\mathbf{x}(t)]|s(t) = i\} \tag{5.24}$$

where the function $\mathcal{Q}[\mathbf{x}(t)]$ vector quantises the observation vector $\mathbf{x}(t)$ to the nearest discrete vector in the set $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \dots, \boldsymbol{\mu}_{iM}]$. For discrete density HMMs, the probability of a state

vector $\boldsymbol{\mu}_{ik}$ can be defined as the ratio of the number of occurrences of $\boldsymbol{\mu}_{ik}$ (or vectors quantised to $\boldsymbol{\mu}_{ik}$) in the state i , divided by the total number of occurrences of all other vectors in the state i :

$$\begin{aligned}\bar{P}_{ik}(\boldsymbol{\mu}_{ik}) &= \frac{\text{expected number of times in state } i \text{ and observing } \boldsymbol{\mu}_{ik}}{\text{expected number of times in state } i} \\ &= \frac{\sum_{t \in \mathbf{x}(t) \rightarrow \boldsymbol{\mu}_{ik}} \gamma_t(i)}{\sum_{t=0}^{T-1} \gamma_t(i)}\end{aligned}\quad (5.25)$$

In Equation (5.25) the summation in the numerator is taken over those time instants t where the k th symbol $\boldsymbol{\mu}_{ik}$ is observed in the state i .

For statistically reliable results, an HMM must be trained on a large data set, \mathbf{X} , consisting of a sufficient number of independent realisations of the process to be modelled. Assume that the training data set consists of L realisations, $\mathbf{X} = [\mathbf{X}(0), \mathbf{X}(1), \dots, \mathbf{X}(L-1)]$, where $\mathbf{X}(k) = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T_k - 1)]$. The re-estimation formula can be averaged over the entire data set as

$$\hat{\pi}_i = \frac{1}{L} \sum_{l=0}^{L-1} \gamma_0^l(i) \quad (5.26)$$

$$\hat{a}_{ij} = \frac{\sum_{l=0}^{L-1} \sum_{t=0}^{T_l-2} \gamma_t^l(i, j)}{\sum_{l=0}^{L-1} \sum_{t=0}^{T_l-2} \gamma_t^l(i)} \quad (5.27)$$

and

$$\hat{P}_i(\boldsymbol{\mu}_{ik}) = \frac{\sum_{l=0}^{L-1} \sum_{t \in \mathbf{x}(t) \rightarrow \boldsymbol{\mu}_{ik}} \gamma_t^l(i)}{\sum_{l=0}^{L-1} \sum_{t=0}^{T_l-1} \gamma_t^l(i)} \quad (5.28)$$

The parameter estimates of Equations (5.26)–(5.28) can be used in further iterations of the estimation process until the model converges.

5.3.4 HMMs WITH CONTINUOUS DENSITY OBSERVATION MODELS

In continuous density HMMs, continuous pdfs are used to model the space of the observation signals associated with each state. Baum *et al.* generalised the parameter re-estimation method to HMMs with concave continuous pdfs such as a Gaussian pdf. A continuous P -variate Gaussian pdf for the state i of an HMM can be defined as

$$f_{\mathbf{x}|s}[\mathbf{x}(t) | s(t) = i] = \frac{1}{(2\pi)^{P/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\{[\mathbf{x}(t) - \boldsymbol{\mu}_i]^T \boldsymbol{\Sigma}_i^{-1} [\mathbf{x}(t) - \boldsymbol{\mu}_i]\} \quad (5.29)$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and the covariance matrix associated with the state i . The re-estimation formula for the mean vector of the state Gaussian pdf can be derived as

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i) \mathbf{x}(t)}{\sum_{t=0}^{T-1} \gamma_t(i)} \quad (5.30)$$

Similarly, the covariance matrix is estimated as

$$\bar{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i) [\mathbf{x}(t) - \bar{\boldsymbol{\mu}}_i] [\mathbf{x}(t) - \bar{\boldsymbol{\mu}}_i]^T}{\sum_{t=0}^{T-1} \gamma_t(i)} \quad (5.31)$$

The proof that the Baum–Welch re-estimation algorithm leads to maximisation of the likelihood function $f_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M})$ can be found in Baum.

5.3.5 HMMs WITH GAUSSIAN MIXTURE PDFs

The modelling of the space of a signal process with a mixture of Gaussian pdfs is considered in Section 4.5. In HMMs with a Gaussian mixture pdf for a state observation model, the signal space associated with the i th state is modelled with a mixture of M Gaussian densities as

$$f_{\mathbf{X}|S}[\mathbf{x}(t) | s(t) = i] = \sum_{k=1}^M P_{ik} \mathcal{N}[\mathbf{x}(t), \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}] \quad (5.32)$$

where P_{ik} is the prior probability of the k th component of the mixture. The posterior probability of state i at time t and state j at time $t + 1$ of the model, \mathcal{M} , given an observation sequence $\mathbf{X} = [\mathbf{x}(0), \dots, \mathbf{x}(T - 1)]$, can be expressed as

$$\begin{aligned} \gamma_t(i, j) &= P_{S|\mathbf{X}, \mathcal{M}}[s(t) = i, s(t+1) = j | \mathbf{X}, \mathcal{M}] \\ &= \frac{\alpha_t(i) a_{ij} \left\{ \sum_{k=1}^M P_{jk} \mathcal{N}[\mathbf{x}(t+1), \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}] \right\} \beta_{t+1}(j)}{\sum_{i=1}^N \alpha_{T-1}(i)} \end{aligned} \quad (5.33)$$

and the posterior probability of state i at time t given the model, \mathcal{M} , and the observation, \mathbf{X} , is given by

$$\begin{aligned} \gamma_t(i) &= P_{S|\mathbf{X}, \mathcal{M}}[s(t) = i | \mathbf{X}, \mathcal{M}] \\ &= \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^N \alpha_{T-1}(j)} \end{aligned} \quad (5.34)$$

Now we define the joint posterior probability of the state i and the k th Gaussian mixture component pdf model of the state i at time t as

$$\begin{aligned} \zeta_t(i, k) &= P_{S, K | \mathbf{X}, \mathcal{M}} [s(t) = i, m(t) = k | \mathbf{X}, \mathcal{M}] \\ &= \frac{\sum_{j=1}^N \alpha_{t-1}(j) a_{ji} P_{ik} \mathcal{N}[\mathbf{x}(t), \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}] \beta_i(i)}{\sum_{j=1}^N \alpha_{t-1}(j)} \end{aligned} \quad (5.35)$$

where $m(t)$ is the Gaussian mixture component at time t . Equations (5.33)–(5.35) are used to derive the re-estimation formula for the mixture coefficients, the mean vectors and the covariance matrices of the state mixture Gaussian pdfs as

$$\begin{aligned} \bar{P}_{ik} &= \frac{\text{expected number of times in state } i \text{ and observing mixture } k}{\text{expected number of times in state } i} \\ &= \frac{\sum_{t=0}^{T-1} \xi_t(i, k)}{\sum_{t=0}^{T-1} \gamma_t(i)} \end{aligned} \quad (5.36)$$

and

$$\bar{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=0}^{T-1} \xi_t(i, k) \mathbf{x}(t)}{\sum_{t=0}^{T-1} \xi_t(i, k)} \quad (5.37)$$

Similarly the covariance matrix is estimated as

$$\bar{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{t=0}^{T-1} \xi_t(i, k) [\mathbf{x}(t) - \bar{\boldsymbol{\mu}}_{ik}] [\mathbf{x}(t) - \bar{\boldsymbol{\mu}}_{ik}]^T}{\sum_{t=0}^{T-1} \xi_t(i, k)} \quad (5.38)$$

5.4 DECODING OF SIGNALS USING HIDDEN MARKOV MODELS

Hidden Markov models are used in applications such as speech recognition, image recognition and signal restoration, and for the decoding of the underlying states of a signal. For example, in speech recognition, HMMs are trained to model the statistical variations of the acoustic realisations of the words in a vocabulary of, say, size V words. In the word recognition phase, an utterance is classified and labelled with the most likely of the $V + 1$ candidate HMMs (including an HMM for silence), as illustrated in Figure 5.10. In Chapter 12 on the modelling and detection of impulsive noise, a binary-state HMM is used to model the impulsive noise process.

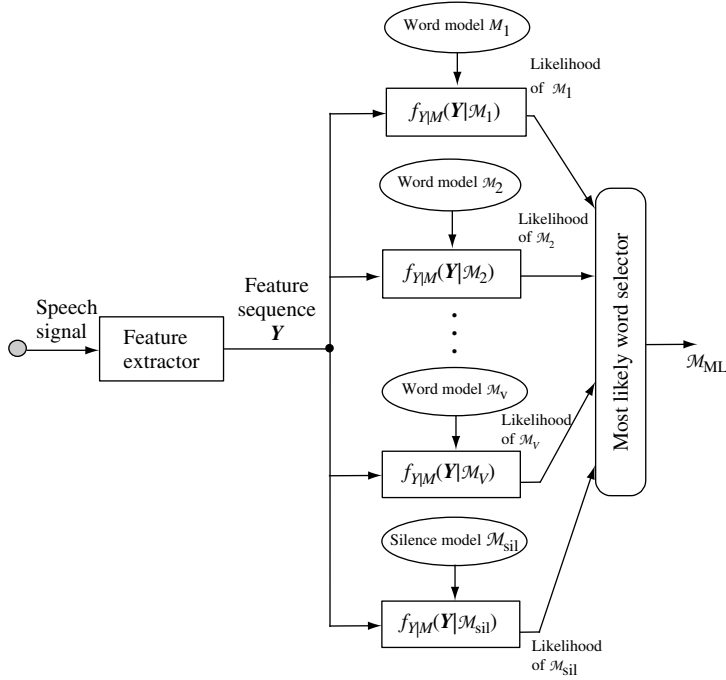


Figure 5.10 Illustration of the use of HMMs in speech recognition.

Consider the decoding of an unlabelled sequence of T signal vector, $\mathbf{X} = [\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T-1)]$, given a set of V candidate HMMs $[\mathcal{M}_1, \dots, \mathcal{M}_V]$. The probability score for the observation vector sequence, \mathbf{X} , and the model, \mathcal{M}_k , can be calculated as the likelihood:

$$f_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M}_k) = \sum_s \pi_{s(0)} f_{\mathbf{X}|S}[\mathbf{x}(0)|s(0)] a_{s(0)s(1)} f_{\mathbf{X}|S}[\mathbf{x}(1)|s(1)] \cdots a_{s(T-2)s(T-1)} f_{\mathbf{X}|S}[\mathbf{x}(T-1)|s(T-1)] \quad (5.39)$$

where the likelihood of the observation sequence, \mathbf{X} , is summed over all possible state sequences of the model \mathcal{M} . Equation (5.39) can be efficiently calculated using the forward-backward method described in Section 5.3.1. The observation sequence \mathbf{X} is labelled with the HMM that scores the highest likelihood as

$$\text{Label}(\mathbf{X}) = \arg \max_k [f_{\mathbf{X}|\mathcal{M}}(\mathbf{X}|\mathcal{M}_k)], \quad k = 1, \dots, V + 1 \quad (5.40)$$

In decoding applications often the likelihood of an observation sequence, \mathbf{X} , and a model, \mathcal{M}_k , is obtained along the *single* most likely state sequence of model \mathcal{M}_k , instead of being summed over all sequences, so Equation (5.40) becomes

$$\text{Label}(\mathbf{X}) = \arg \max_k \left[\max_s f_{\mathbf{X},S|\mathcal{M}}(\mathbf{X}, \mathbf{s}|\mathcal{M}_k) \right] \quad (5.41)$$

In Section 5.5, on the use of HMMs for noise reduction, the most likely state sequence is used to obtain the maximum-likelihood estimate of the underlying statistics of the signal process.

5.4.1 VITERBI DECODING ALGORITHM

In this section, we consider the decoding of a signal to obtain the MAP estimate of the underlying state sequence. The MAP state sequence s^{MAP} of a model, \mathcal{M} , given an observation signal sequence, $\mathbf{X} = [\mathbf{x}(0), \dots, \mathbf{x}(T - 1)]$, is obtained as

$$\begin{aligned} s^{\text{MAP}} &= \arg \max_s f_{\mathbf{X},s|\mathcal{M}}(\mathbf{X}, s | \mathcal{M}) \\ &= \arg \max_s [f_{\mathbf{X}|s,\mathcal{M}}(\mathbf{X} | s, \mathcal{M}) P_{s|\mathcal{M}}(s | \mathcal{M})] \end{aligned} \tag{5.42}$$

The MAP state sequence estimate is used in such applications as the calculation of a similarity score between a signal sequence, \mathbf{X} , and an HMM, \mathcal{M} , segmentation of a nonstationary signal into a number of distinct quasistationary segments, and implementation of state-based Wiener filters for restoration of noisy signals, as described in the next section.

For an N -state HMM and an observation sequence of length T , there are altogether N^T state sequences. Even for moderate values of N and T (say $N = 6$ and $T = 30$), an exhaustive search of the state–time trellis for the best state sequence is a computationally prohibitive exercise. The Viterbi algorithm is an efficient method for the estimation of the most likely state sequence of an HMM. In a state–time trellis diagram, such as Figure 5.8, the number of paths diverging from each state of a trellis can grow exponentially by a factor of N at successive time instants. The Viterbi method prunes the trellis by selecting the most likely path to each state. At each time instant t , for each state i , the algorithm selects the most probable path to state i and prunes out the less likely branches. This procedure ensures that, at any time instant, only a single path *survives* into each state of the trellis.

For each time instant t and for each state i , the algorithm keeps a record of the state j from which the maximum-likelihood path branched into i , and also records the cumulative probability of the most likely path into state i at time t . The Viterbi algorithm is given on the next page, and Figure 5.11 gives a network illustration of the algorithm.

The backtracking routine retrieves the most likely state sequence of the model \mathcal{M} . Note that the variable $Prob_{\text{max}}$, which is the probability of the observation sequence

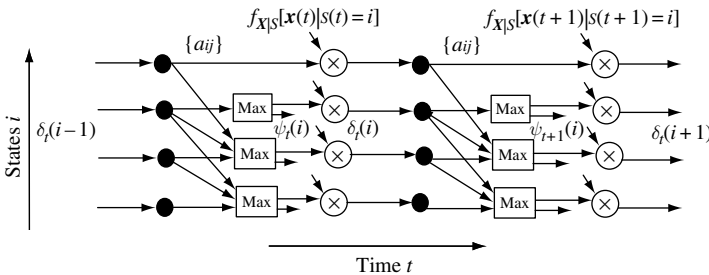


Figure 5.11 A network illustration of the Viterbi algorithm.

$X = [x(0), \dots, x(T - 1)]$ and the most likely state sequence of the model \mathcal{M} , can be used as the probability score for the model \mathcal{M} and the observation X . For example, in speech recognition, for each candidate word model the probability of the observation and the most likely state sequence are calculated, and then the observation is labelled with the word that achieves the highest probability score.

5.5 HMMs IN DNA AND PROTEIN SEQUENCE MODELLING

A major application of hidden Markov models is in biosignal processing and computational molecular biology in applications including multiple alignment and functional classification of proteins, prediction of protein folding, recognition of genes in bacterial and human genomes, analysis and prediction of DNA functional sites and identification of nucleosomal DNA periodical patterns.

Hidden Markov models are powerful probabilistic models for detecting homology among evolutionarily related sequences. Homology is concerned with likeness in structures between parts of different organisms due to evolutionary differentiation from the same or a corresponding part of a remote ancestor.

HMMs are statistical models that consider all possible combinations of matches, mismatches and gaps to generate an alignment of a set of sequences. Figure 5.12 shows a simple example of statistical modeling of DNA observations. In this case the observations are nucleotides and the aim of modelling is to align and estimate the sequential probabilities of observation sequences composed of DNA labels ACGT. Each row shows a DNA sequences. Each column is a state for which the probabilities of occurrence of ACTG are calculated as the normalized number of occurrences of each letter in the column.

Figure 5.13 shows a widely used profile-HMM structure for DNA and protein sequencing. HMMs that represent a sequence profile of a group of related proteins or DNAs are *profile*

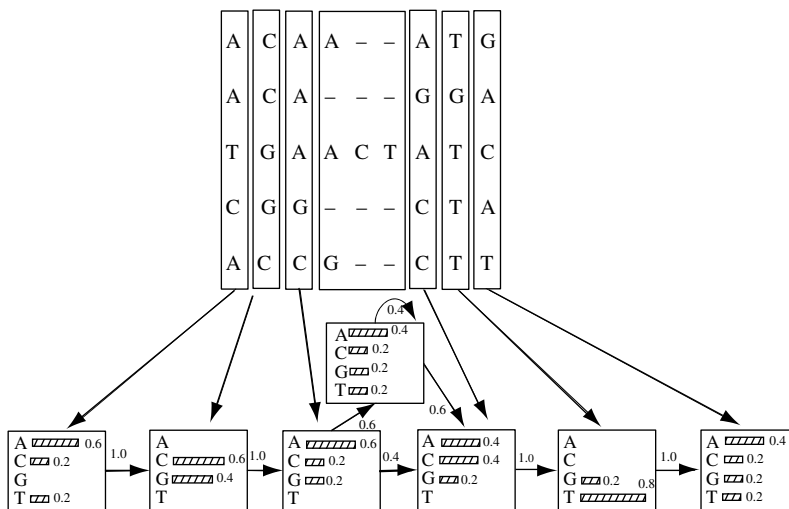


Figure 5.12 A Markov model for a dataset of DNA sequences. The discrete probabilities are histograms of occurrence of each symbol in a column.

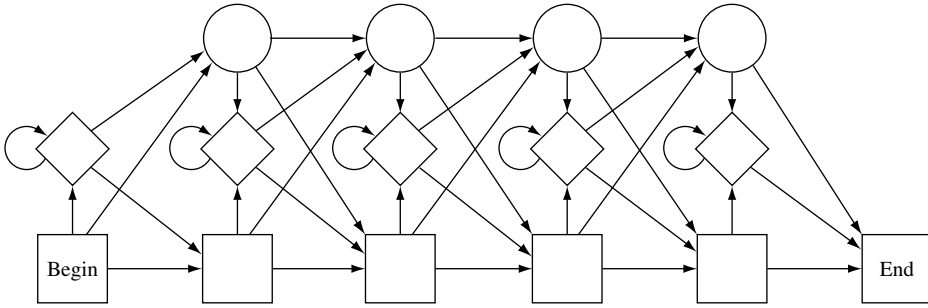


Figure 5.13 A DNA profile HMM: squares represent base states, diamonds are insertion states and circles are deletion states.

HMMs. Again, squares represent main states, diamonds are insertion states and circles are deletion states. There are three possible ‘states’ for each amino acid position in a particular sequence alignment: a ‘main’ state where an amino acid can match or mismatch, an ‘insert’ state where a new amino acid can be added to one of the sequences to generate an alignment, or a ‘delete’ state where an amino acid can be deleted from one of the sequences to generate the alignment. Probabilities are assigned to each of these states based on the number of each of these events encountered in the sequence alignment. An arrow in the model represents a transition from one state to another and is also associated with a transition probability. The greater the number and diversity of sequences included in the training alignment, the better the model will be at identifying related sequences.

An adequately ‘trained’ profile HMM has many uses. It can align a group of related sequences, search databases for distantly related sequences, and identify subfamily-specific signatures within large protein or DNA superfamilies.

5.6 HMMs FOR MODELLING SPEECH AND NOISE

5.6.1 MODELLING SPEECH WITH HMMs

HMMs are the main statistical modelling framework for speech recognition. Normally a three-to-five-states HMM, with 10–20 Gaussian mixture pdfs per state, is used to model the statistical variations of the spectral and temporal features of a phonemic unit of speech. Each state of an HMM of a phoneme models a sub-phonemic segment with the first state modelling the first segment of the phoneme and the second state modelling the second segment and so on. Speech recognition is described in Chapter 11. For implementation of HMMs of speech, the hidden Markov model toolkit (HTK) provides a good platform.

5.6.2 HMM-BASED ESTIMATION OF SIGNALS IN NOISE

In this section, and the following two sections, we consider the use of HMMs for estimation of a signal $x(t)$ observed in an additive noise $n(t)$, and modelled as

$$y(t) = x(t) + n(t) \quad (5.43)$$

From Bayes's rule, the posterior pdf of the signal $\mathbf{x}(t)$, given the noisy observation $\mathbf{y}(t)$, is defined as

$$\begin{aligned} f_{\mathbf{x}|\mathbf{y}}[\mathbf{x}(t)|\mathbf{y}(t)] &= \frac{f_{\mathbf{y}|\mathbf{x}}[\mathbf{y}(t)|\mathbf{x}(t)]f_X[\mathbf{x}(t)]}{f_Y[\mathbf{y}(t)]} \\ &= \frac{1}{f_Y[\mathbf{y}(t)]}f_N[\mathbf{y}(t)-\mathbf{x}(t)]f_X[\mathbf{x}(t)] \end{aligned} \quad (5.44)$$

For a given observation, $f_Y[\mathbf{y}(t)]$ is a constant, and the MAP estimate is obtained as

$$\hat{\mathbf{x}}^{\text{MAP}}(t) = \arg \max_{\mathbf{x}(t)} f_N[\mathbf{y}(t)-\mathbf{x}(t)]f_X[\mathbf{x}(t)] \quad (5.45)$$

The computation of the posterior pdf, Equation (5.44), or the MAP estimate Equation (5.45), requires the pdf models of the signal and the noise processes. Stationary, continuous-value processes are often modelled by a Gaussian or a mixture Gaussian pdf that is equivalent to a single-state HMM. For a nonstationary process an N -state HMM can model the time-varying pdf of the process as a Markovian chain of N stationary Gaussian subprocesses. Now assume that we have an N_s -state HMM \mathcal{M} for the signal, and another N_n -state HMM η for the noise. For signal estimation, we need estimates of the underlying state sequences of the signal and the noise processes. For an observation sequence of length T , there are N_s^T possible signal state sequences and N_n^T possible noise state sequences that could have generated the noisy signal. Since it is assumed that the signal and noise are uncorrelated, each signal state may be observed in any noisy state; therefore the number of noisy signal states is on the order of $N_s^T \times N_n^T$.

Given an observation sequence $\mathbf{Y} = [\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(T-1)]$, the most probable state sequences of the signal and the noise HMMs may be expressed as

$$\mathbf{s}_{\text{signal}}^{\text{MAP}} = \arg \max_{\mathbf{s}_{\text{signal}}} \left[\max_{\mathbf{s}_{\text{noise}}} f_Y(\mathbf{Y}, \mathbf{s}_{\text{signal}}, \mathbf{s}_{\text{noise}} | \mathcal{M}, \eta) \right] \quad (5.46)$$

and

$$\mathbf{s}_{\text{noise}}^{\text{MAP}} = \arg \max_{\mathbf{s}_{\text{noise}}} \left[\max_{\mathbf{s}_{\text{signal}}} f_Y(\mathbf{Y}, \mathbf{s}_{\text{signal}}, \mathbf{s}_{\text{noise}} | \mathcal{M}, \eta) \right] \quad (5.47)$$

Given the state sequence estimates for the signal and the noise models, the MAP estimation, Equation (5.45), becomes

$$\hat{\mathbf{x}}^{\text{MAP}}(t) = \arg \max_{\mathbf{x}} \{ f_{N|\mathcal{S},\eta}[\mathbf{y}(t)-\mathbf{x}(t)|\mathbf{s}_{\text{noise}}^{\text{MAP}}, \eta] f_{X|\mathcal{S},\mathcal{M}}[\mathbf{x}(t)|\mathbf{s}_{\text{signal}}^{\text{MAP}}, \mathcal{M}] \} \quad (5.48)$$

Implementation of Equations (5.46)–(5.48) is computationally prohibitive. In Sections 5.6.4 and 5.6.5, we consider some practical methods for the estimation of signal in noise.

Example 5.1

Assume a signal, modelled by a binary-state HMM, is observed in an additive stationary Gaussian noise. Let the noisy observation be modelled as

$$\mathbf{y}(t) = \bar{s}(t)\mathbf{x}_0(t) + s(t)\mathbf{x}_1(t) + \mathbf{n}(t) \quad (5.49)$$

where $s(t)$ is a hidden binary-state process such that: $s(t) = 0$ indicates that the signal is from the state S_0 with a Gaussian pdf of $\mathcal{N}[\mathbf{x}(t), \boldsymbol{\mu}_{x_0}, \boldsymbol{\Sigma}_{x_0x_0}]$, and $s(t) = 1$ indicates that the signal is from the state S_1 with a Gaussian pdf of $\mathcal{N}[\mathbf{x}(t), \boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1x_1}]$. Assume that a stationary Gaussian process, $\mathcal{N}[\mathbf{n}(t), \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_{nn}]$, equivalent to a single-state HMM, can model the noise. Using the Viterbi algorithm, the MAP state sequence of the signal model can be estimated as

$$s_{\text{signal}}^{\text{MAP}} = \arg \max_s [f_{Y|S, \mathcal{M}}(\mathbf{Y}|s, \mathcal{M}) P_{S|\mathcal{M}}(s|\mathcal{M})] \quad (5.50)$$

For a Gaussian-distributed signal and additive Gaussian noise, the observation pdf of the noisy signal is also Gaussian. Hence, the state observation pdfs of the signal model can be modified to account for the additive noise as

$$f_{Y|s_0}[y(t)|s_0] = \mathcal{N}[y(t), (\boldsymbol{\mu}_{x_0} + \boldsymbol{\mu}_n), (\boldsymbol{\Sigma}_{x_0x_0} + \boldsymbol{\Sigma}_{nn})] \quad (5.51)$$

and

$$f_{Y|s_1}[y(t)|s_1] = \mathcal{N}[y(t), (\boldsymbol{\mu}_{x_1} + \boldsymbol{\mu}_n), (\boldsymbol{\Sigma}_{x_1x_1} + \boldsymbol{\Sigma}_{nn})] \quad (5.52)$$

where $\mathcal{N}[y(t), \boldsymbol{\mu}, \boldsymbol{\Sigma}]$ denotes a Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The MAP signal estimate, given a state sequence estimate s^{MAP} , is obtained from

$$\hat{\mathbf{x}}^{\text{MAP}}(t) = \arg \max_{\mathbf{x}} \{f_{X|S, \mathcal{M}}[\mathbf{x}(t)|s^{\text{MAP}}, \mathcal{M}] f_N[y(t) - \mathbf{x}(t)]\} \quad (5.53)$$

Substitution of the Gaussian pdf of the signal from the most likely state sequence, and the pdf of noise, in Equation (5.53), results in the following MAP estimate:

$$\hat{\mathbf{x}}^{\text{MAP}}(t) = (\boldsymbol{\Sigma}_{xx, s(t)} + \boldsymbol{\Sigma}_{nn})^{-1} \boldsymbol{\Sigma}_{xx, s(t)} [y(t) - \boldsymbol{\mu}_n] + (\boldsymbol{\Sigma}_{xx, s(t)} + \boldsymbol{\Sigma}_{nn})^{-1} \boldsymbol{\Sigma}_{nn} \boldsymbol{\mu}_{x, s(t)} \quad (5.54)$$

where $\boldsymbol{\mu}_{x, s(t)}$ and $\boldsymbol{\Sigma}_{xx, s(t)}$ are the mean vector and covariance matrix of the signal $\mathbf{x}(t)$ obtained from the most likely state sequence, $[s(t)]$.

5.6.3 SIGNAL AND NOISE MODEL COMBINATION AND DECOMPOSITION

For Bayesian estimation of a signal observed in additive noise, we need to have an estimate of the underlying statistical state sequences of the signal and the noise processes. Figure 5.14 illustrates the outline of an HMM-based noisy speech recognition and enhancement system. The system performs the following functions:

- (1) combination of the speech and noise HMMs to form the noisy speech HMMs;
- (2) estimation of the best combined noisy speech model given the current noisy speech input;
- (3) state decomposition, i.e. the separation of speech and noise states given noisy speech states;
- (4) state-based Wiener filtering using the estimates of speech and noise states.

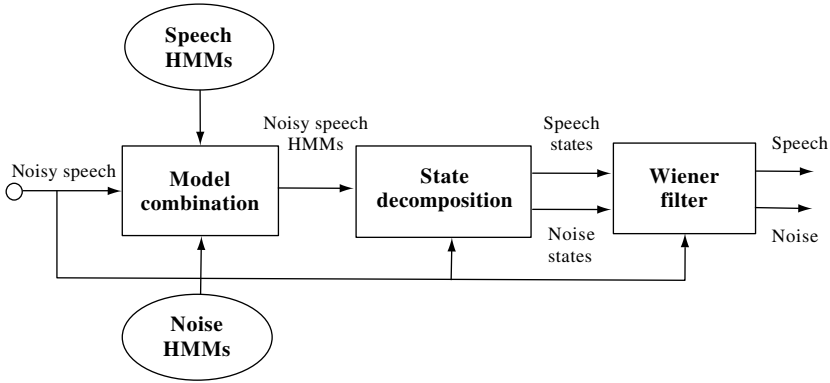


Figure 5.14 Outline configuration of HMM-based noisy speech recognition and enhancement.

5.6.4 HIDDEN MARKOV MODEL COMBINATION

The performance of HMMs trained on clean signals deteriorates rapidly in the presence of noise, since noise causes a mismatch between the clean HMMs and the noisy signals. The noise-induced mismatch can be reduced, either by filtering the noise from the signal (for example using the Wiener filtering and the spectral subtraction methods described in Chapters 6 and 11) or by combining the noise and the signal models to model the noisy signal.

The model combination method, illustrated in Figure 5.15, was developed by Gales and Young. In this method HMMs of speech are combined with an HMM of noise to form

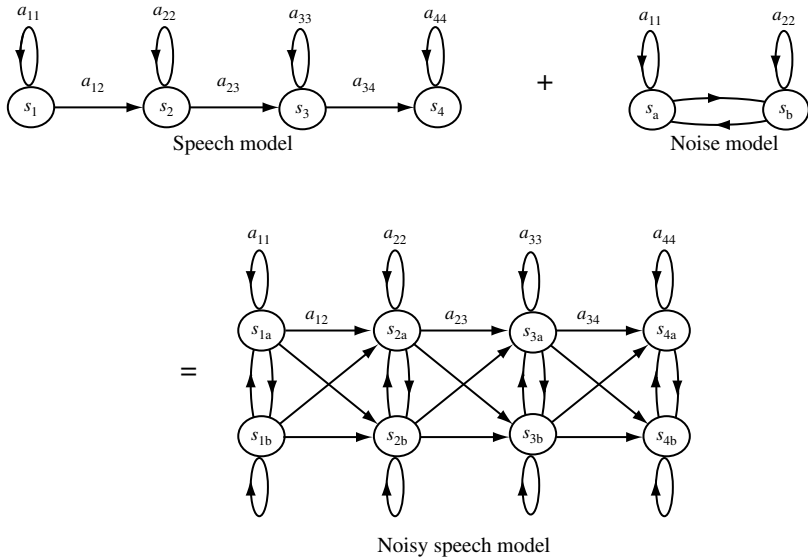


Figure 5.15 Outline configuration of HMM-based noisy speech recognition and enhancement. s_{ij} is a combination of the state i of speech with the state j of noise.

HMMs of noisy speech signals. In the power-spectral domain, the mean vector and the covariance matrix of the noisy speech can be approximated by adding the mean vectors and the covariance matrices of speech and noise models:

$$\boldsymbol{\mu}_y = \boldsymbol{\mu}_x + g\boldsymbol{\mu}_n \quad (5.55)$$

$$\boldsymbol{\Sigma}_{yy} = \boldsymbol{\Sigma}_{xx} + g^2\boldsymbol{\Sigma}_{nn} \quad (5.56)$$

Model combination also requires an estimate of the current signal-to-noise ratio for calculation of the scaling factor g in Equations (5.55) and (5.56). In cases such as speech recognition, where the models are trained on cepstral features, the model parameters are first transformed from cepstral features into power spectral features before using the additive linear combination Equations (5.55) and (5.56). Figure 5.15 illustrates the combination of a four-state left–right HMM of a speech signal with a two-state ergodic HMM of noise. Assuming that speech and noise are independent processes, each speech state must be combined with every possible noise state to give the noisy speech model. It is assumed that the noise process only affects the mean vectors and the covariance matrices of the speech model; hence the transition probabilities of the speech model are not modified.

5.6.5 DECOMPOSITION OF STATE SEQUENCES OF SIGNAL AND NOISE

The HMM-based state decomposition problem can be stated as follows: given a noisy signal and the HMMs of the signal and the noise processes, estimate the underlying states of the signal and the noise. HMM state decomposition can be obtained using the following method:

- (1) given the noisy signal and a set of combined signal and noise models, estimate the maximum-likelihood (ML) combined noisy HMM for the noisy signal;
- (2) obtain the ML state sequence of from the ML combined model;
- (3) extract the signal and noise states from the ML state sequence of the ML combined noisy signal model.

The ML state sequences provide the probability density functions for the signal and noise processes. The ML estimates of the speech and noise pdfs may then be used in Equation (5.45) to obtain a MAP estimate of the speech signal. Alternatively the mean spectral vectors of the speech and noise from the ML state sequences can be used to program a state-dependent Wiener filter, as described in the next section.

5.6.6 HMM-BASED WIENER FILTERS

The least mean square error Wiener filter is derived in Chapter 6. For a stationary signal $x(m)$, observed in an additive noise $n(m)$, the Wiener filter equations in the time and the frequency domains are derived as:

$$\boldsymbol{w} = (\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})^{-1}\boldsymbol{r}_{xx} \quad (5.57)$$

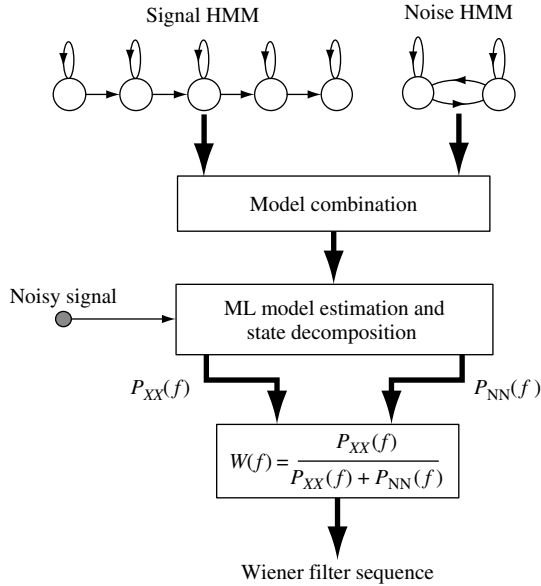


Figure 5.16 Illustrations of HMMs with state-dependent Wiener filters.

and

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \quad (5.58)$$

where \mathbf{R}_{xx} , \mathbf{r}_{xx} and $P_{XX}(f)$ denote the autocorrelation matrix, the autocorrelation vector and the power-spectral functions, respectively. The implementation of the Wiener filter, Equation (5.58), requires the signal and the noise power spectra. The power-spectral variables may be obtained from the ML states of the HMMs trained to model the power spectra of the signal and the noise. Figure 5.16 illustrates an implementation of HMM-based state-dependent Wiener filters. To implement the state-dependent Wiener filter, we need an estimate of the state sequences for the signal and the noise. In practice, for signals such as speech there are a number of HMMs; one HMM per word, phoneme or any other elementary unit of the signal. In such cases it is necessary to classify the signal, so that the state-based Wiener filters are derived from the most likely HMM. Furthermore the noise process can also be modelled by an HMM. Assuming that there are V HMMs $\{\mathcal{M}_1, \dots, \mathcal{M}_V\}$ for the signal process, and one HMM for the noise, the state-based Wiener filter can be implemented as follows:

- Step 1:* combine the signal and noise models to form the noisy signal models.
- Step 2:* given the noisy signal, and the set of combined noisy signal models, obtain the ML combined noisy signal model.
- Step 3:* from the ML combined model, obtain the ML state sequence of speech and noise.
- Step 4:* use the ML estimate of the power spectra of the signal and the noise to program the Wiener filter, Equation (5.56).
- Step 5:* use the state-dependent Wiener filters to filter the signal.

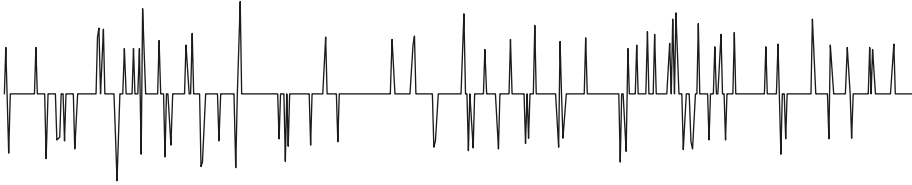


Figure 5.17 Impulsive noise.

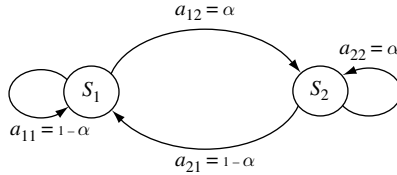


Figure 5.18 A binary-state model of an impulsive noise process.

5.6.7 MODELLING NOISE CHARACTERISTICS

The implicit assumption in using an HMM for noise is that noise statistics can be modelled by a Markovian chain of N different stationary processes. A stationary noise process can be modelled by a single-state HMM. For a nonstationary noise, a multistate HMM can model the time variations of the noise process with a finite number of quasistationary states. In general, the number of states required to accurately model the noise depends on the nonstationary character of the noise.

An example of a nonstationary noise process is the impulsive noise of Figure 5.17. Figure 5.18 shows a two-state HMM of the impulsive noise sequence where the state S_0 models the ‘off’ periods between the impulses and the state S_1 models an impulse. In cases where each impulse has a well-defined temporal structure, it may be beneficial to use a multistate HMM to model the pulse itself. HMMs are used in Chapter 12 for modelling impulsive noise, and in Chapter 15 for channel equalisation.

5.7 SUMMARY

HMMs provide a powerful method for the modelling of nonstationary processes such as speech, noise and time-varying channels. An HMM is a Bayesian finite-state process, with a Markovian state prior, and a state likelihood function that can be either a discrete density model or a continuous Gaussian pdf model. The Markovian prior models the time evolution of a nonstationary process with a chain of stationary subprocesses. The state observation likelihood models the space of the process within each state of the HMM.

In Section 5.3, we studied the Baum–Welch method for the training of the parameters of an HMM to model a given data set, and derived the forward–backward method for efficient calculation of the likelihood of an HMM given an observation signal. In Section 5.4, we considered the use of HMMs in signal classification and in the decoding of the underlying

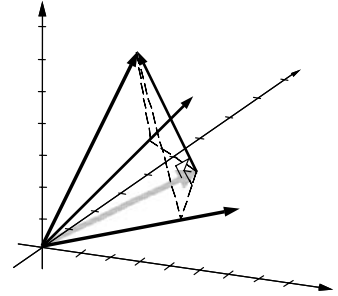
state sequence of a signal. The Viterbi algorithm is a computationally efficient method for estimation of the most likely sequence of an HMM. Given an unlabelled observation signal, the decoding of the underlying state sequence and the labelling of the observation with one of number of candidate HMMs are accomplished using the Viterbi method. In Section 5.5, we considered the use of HMMs for MAP estimation of a signal observed in noise, and considered the use of HMMs in implementation of the state-based Wiener filter sequence.

BIBLIOGRAPHY

- Bahl L.R., Jelinek F. and Mercer R.L. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **5**: 179–190.
- Bahl L.R., Brown P.F., de Souza P.V. and Mercer R.L. (1986) Maximum mutual information estimation of hidden Markov model parameters for speech recognition. *IEEE Proceedings, Acoustics, Speech and Signal Processing, ICASSP-86*, Tokyo, pp. 40–43.
- Baum L.E. and Eagon J.E. (1967) An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to models for ecology. *Bull. AMS*, **73**: 360–363.
- Baum L.E. and Petrie T. (1966) Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Statist.*, **37**: 1554–1563.
- Baum L.E., Petrie T., Soules G. and Weiss N. (1970) A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, **41**: 164–171.
- Conner P.N. (1993) Hidden Markov model with improved observation and duration modelling. Ph.D. thesis, University of East Anglia, UK.
- Epharaim Y., Malah D. and Juang B.H. (1989) On application of Hidden Markov models for enhancing noisy speech. *IEEE Trans. Acoust. Speech Signal Process.*, **37**(12): 1846–1856.
- Forney G.D. (1973) The Viterbi algorithm. *Proc. IEEE*, **61**: 268–278.
- Gales M.J.F. and Young S.J. (1992) An improved approach to the hidden Markov model decomposition of speech and noise. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-92*, pp. 233–235.
- Gales M.J.F. and Young S.J. (1993) HMM recognition in noise using parallel model combination. *Eurospeech-93*, pp. 837–840.
- Huang X.D. and Jack M.A. (1989) Unified techniques for vector quantisation and hidden Markov modelling using semi-continuous models. *IEEE Proceedings, Acoustics, Speech and Signal Processing, ICASSP-89*, Glasgow, pp. 639–642.
- Huang X.D., Ariki Y. and Jack M.A. (1990) *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, Edinburgh.
- Jelinek F. (1976) Continuous speech recognition by statistical methods. *Proc. IEEE*, **64**: 532–555.
- Jelinek F. and Mercer R. (1980) Interpolated estimation of Markov source parameters from sparse data. *Proceedings of the Workshop on Pattern Recognition in Practice*. North-Holland, Amsterdam.
- Juang B.H. (1984) On the hidden Markov model and dynamic time warping for speech recognition – a unified overview. *AT&T Bell Lab. Tech. J.*, **63**: 1213–1243.
- Juang B.H. (1985) Maximum-likelihood estimation for mixture multi-variate stochastic observations of Markov chain. *AT&T Bell Lab. Tech. J.*, **64**: 1235–1249.
- Kullback S. and Leibler R.A. (1951) On information and sufficiency. *Ann. Math. Statist.*, **22**: 79–85.
- Lee K.F. (1989a) *Automatic Speech Recognition: the Development of SPHINX System*. Kluwer Academic, Boston, MA.
- Lee K.F. (1989b) Hidden Markov model: past, present and future. *Eurospeech-89*, Paris.
- Liporace L.R. (1982) Maximum likelihood estimation for multi-variate observations of Markov sources. *IEEE Trans. IT*, **IT-28**: 729–735.
- Markov A.A. (1913) An example of statistical investigation in the text of *Eugen Onyegin* illustrating coupling of tests in chains. *Proc. Acad. Sci. St Petersburg VI Ser.*, **7**: 153–162.
- Milner B.P. (1995) Speech recognition in adverse environments. Ph.D. Thesis, University of East Anglia, UK.
- Petierie T. (1969) Probabilistic functions of finite state Markov chains. *Ann. Math. Statist.*, **40**: 97–115.
- Rabiner L.R. and Juang B.H. (1986) An introduction to hidden Markov models. *IEEE ASSP. Mag.*, pp. 4–15.

- Rabiner L.R. and Juang B.H. (1993) *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ.
- Rabiner L.R., Juang B.H., Levinson S.E. and Sondhi M.M., (1985) Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Bell Lab. Tech. J.*, **64**: 1211–1235.
- Varga A. and Moore R.K. (1990) Hidden Markov model decomposition of speech and noise. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 845–848.
- Viterbi A.J. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, **IT-13**: 260–226.
- Young S.J. (1999) *HTK: Hidden Markov Model Tool Kit*. Cambridge University Engineering Department.

6



Least Square Error Filters

6.1 Least Square Error Estimation: Wiener Filters	166
6.2 Block-data Formulation of the Wiener Filter	170
6.3 Interpretation of Wiener Filters as Projections in Vector Space	172
6.4 Analysis of the Least Mean Square Error Signal	174
6.5 Formulation of Wiener Filters in the Frequency Domain	175
6.6 Some Applications of Wiener Filters	177
6.7 Implementation of Wiener Filters	182
6.8 Summary	185
Bibliography	185

Least square error filter theory, formulated by Norbert Wiener and Andrei Kolmogorov in the 1940s, forms the foundation of data-dependent linear filters. Least square error filters play a central role in a wide range of applications such as linear prediction, echo cancellation, signal restoration, channel equalisation, radar and system identification.

The coefficients of a least square error filter are calculated to minimise the average squared distance between the filter output and a desired or target signal. In its basic form, the least square error filter theory assumes that the signals are stationary processes. However, if the filter coefficients are periodically recalculated and updated for every block of N signal samples, then the filter adapts itself to the average characteristics of the signals within the blocks and becomes block-adaptive. A block-adaptive (or segment-adaptive) filter can be used for signals such as speech and image that may be considered as almost stationary over a relatively small block of samples. In this chapter, we study the least square error filter theory, and consider alternative methods of formulation of the filtering problem. We consider the application of the least square error filters in channel equalisation, time-delay estimation and

additive noise reduction. A case study of the frequency response of the least square error filter, for additive noise reduction, provides useful insight into the operation of the filter. We also deal with some implementation issues of filters.

6.1 LEAST SQUARE ERROR ESTIMATION: WIENER FILTERS

Norbert Wiener, and independently Andrei N. Kolmogorov, formulated the continuous-time least-mean-square error estimation problem of smoothing, interpolation and prediction of signals. Wiener’s work is described in his classic work on interpolation, extrapolation and smoothing of time series (Wiener 1949). The extension of the Wiener filter theory from continuous time to discrete time is simple, and of more practical use for implementation on digital signal processors.

The typical scenario in which a Wiener filter is used is in the context of estimation or prediction of a signal observed in noise. The Wiener filter can be used for signal enhancement to remove the effect of linear distortions such as the de-blurring of distorted or unfocused images or equalisation of the distortion of a telecommunications channel, or noise reduction. A Wiener filter can also be used to predict the trajectory of a projectile; a problem during the Second World War on which Wiener worked. Predicting the fluctuations of a signal from its past values has a wide range of applications from speech and video coding to economic data analysis. The Wiener filter formulation is the basis of least square error applications such as linear prediction and adaptive filters.

A Wiener filter can be an infinite-duration impulse response (IIR) or a finite-duration impulse response (FIR) filter. In this chapter, we consider FIR Wiener filters, since they are relatively simple to compute, inherently stable and more practical. The main drawback of FIR filters compared with IIR filters is that they may need a large number of coefficients to approximate a desired response.

Figure 6.1 illustrates a Wiener filter represented by the filter’s coefficient vector, w . The filter takes as the input a signal $y(m)$, usually a distorted version of a desired signal $x(m)$,

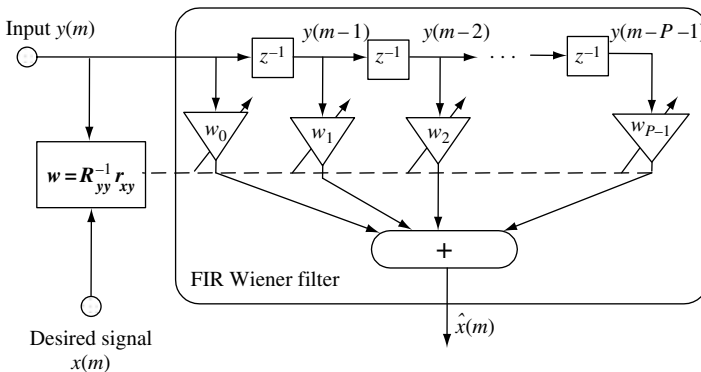


Figure 6.1 Illustration of a Wiener filter structure.

and produces an output signal $\hat{x}(m)$, where $\hat{x}(m)$ is the least-mean-square error estimate of the desired or target signal, $x(m)$. The filter input–output relation is given by

$$\begin{aligned}\hat{x}(m) &= \sum_{k=0}^{P-1} w_k y(m-k) \\ &= \mathbf{w}^T \mathbf{y}\end{aligned}\quad (6.1)$$

where m is the discrete-time index, vector $\mathbf{y}^T = [y(m), y(m-1), \dots, y(m-P+1)]$ is the filter input signal, $\hat{x}(m)$ is the filter output and the parameter vector $\mathbf{w}^T = [w_0, w_1, \dots, w_{P-1}]$ is the Wiener filter coefficient vector. In Equation (6.1), the filtering operation is expressed in two alternative and equivalent representations of a convolutional sum and an inner vector product.

The Wiener filter error signal, $e(m)$, is defined as the difference between the desired (or target) signal, $x(m)$, and the filter output, $\hat{x}(m)$:

$$\begin{aligned}e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \mathbf{w}^T \mathbf{y}\end{aligned}\quad (6.2)$$

where, as expressed in Equation (6.1), $\hat{x}(m)$ is the convolution of the input signal vector, \mathbf{y} , and Wiener filter, \mathbf{w} . In Equation (6.2), for a given input signal, $y(m)$, and a desired signal, $x(m)$, the filter error, $e(m)$, depends on the filter coefficient vector, \mathbf{w} . The Wiener filter is the best filter in the sense of minimising the mean-squared error signal.

To explore the relation between the filter coefficient vector, \mathbf{w} , and the error signal, $e(m)$, we write Equation (6.2) N times in a matrix for a segment of N samples of the signals $[x(0), x(1), \dots, x(N-1)]$ and signals $[y(0), y(1), \dots, y(N-1)]$ as

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} y(0) & y(-1) & y(-2) & \dots & y(1-P) \\ y(1) & y(0) & y(-1) & \dots & y(2-P) \\ y(2) & y(1) & y(0) & \dots & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & y(N-3) & \dots & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix}\quad (6.3)$$

In a compact vector notation this matrix equation may be written as

$$\mathbf{e} = \mathbf{x} - \mathbf{Y}\mathbf{w}\quad (6.4)$$

where \mathbf{e} is the error vector, \mathbf{x} is the desired signal vector, \mathbf{Y} is the input signal matrix and $\mathbf{Y}\mathbf{w} = \hat{\mathbf{x}}$ is the Wiener filter output signal vector. It is assumed that the P initial input signal samples $[y(-1), \dots, y(-P+1)]$ are either known or set to zero.

At this point we explore the dependency of the solution of Equation (6.3) on the number of available samples N , which is also the number of linear equations in Equation (6.3). In Equation (6.3), if the number of given signal samples is equal to the number of unknown filter coefficients, $N = P$, then we have a *square* matrix equation, with as many equations as

there are unknowns, and theoretically there is a unique filter solution \mathbf{w} , with a zero estimate on error $\mathbf{e} = \mathbf{0}$, such that $\hat{\mathbf{x}} = \mathbf{Y}\mathbf{w} = \mathbf{x}$.

If $N < P$ then the number of signal samples N , and hence the number of linear equations, is insufficient to obtain a unique solution for the filter coefficients, in this case there are an infinite number of solutions with zero estimation error, and the matrix equation is said to be *underdetermined*.

In practice, there are two issues: (i) the target signal $x(m)$ is not available; and (ii) the number of signal samples is larger than the filter length. When $N > P$ the matrix equation is said to be *overdetermined* and has a unique solution, usually with a nonzero error. When $N > P$, the filter coefficients are calculated to minimise an average error cost function, such as the mean square error $\mathcal{E}[e^2(m)]$, or the average absolute value of error $\mathcal{E}[|e(m)|]$, where $\mathcal{E}[\cdot]$ is the expectation (averaging) operator. The choice of the error function affects the optimality and the computational complexity of the solution.

In Wiener theory, the objective criterion is the least mean square error (LSE) between the filter output and the desired signal. The least square error criterion is optimal for Gaussian distributed signals. As shown in the following, for FIR filters the LSE criterion leads to linear and closed-form solutions. The Wiener filter coefficients are obtained by minimising an average squared error function, $\mathcal{E}[e^2(m)]$, with respect to the filter coefficient vector, \mathbf{w} , where \mathcal{E} is expectation or average. From Equation (6.2), the mean square estimation error is given by

$$\begin{aligned}\mathcal{E}[e^2(m)] &= \mathcal{E}\{[x(m) - \mathbf{w}^T \mathbf{y}]^2\} \\ &= \mathcal{E}[x^2(m)] - 2\mathbf{w}^T \mathcal{E}[\mathbf{y}x(m)] + \mathbf{w}^T \mathcal{E}[\mathbf{y}\mathbf{y}^T] \mathbf{w} \\ &= \mathbf{r}_{xx}(0) - 2\mathbf{w}^T \mathbf{r}_{yx} + \mathbf{w}^T \mathbf{R}_{yy} \mathbf{w}\end{aligned}\quad (6.5)$$

where $\mathbf{R}_{yy} = \mathcal{E}[\mathbf{y}(m)\mathbf{y}^T(m)]$ is the autocorrelation matrix of the input signal and $\mathbf{r}_{xy} = \mathcal{E}[x(m)\mathbf{y}(m)]$ is the cross-correlation vector of the input and the desired signals. An expanded form of Equation (6.5) can be obtained as

$$\mathcal{E}[e^2(m)] = \mathbf{r}_{xx}(0) - 2 \sum_{k=0}^{P-1} w_k \mathbf{r}_{yx}(k) + \sum_{k=0}^{P-1} w_k \sum_{j=0}^{P-1} w_j \mathbf{r}_{yy}(k-j) \quad (6.6)$$

where $\mathbf{r}_{yy}(k)$ and $\mathbf{r}_{yx}(k)$ are the elements of the autocorrelation matrix \mathbf{R}_{yy} and the cross-correlation vector \mathbf{r}_{xy} , respectively.

From Equation (6.5), the mean square error for an FIR filter is a quadratic function of the filter coefficient vector \mathbf{w} and has a single minimum point. For example, for a filter with two coefficients (w_0, w_1), the mean square error function is a bowl-shaped surface, with a single minimum point, as illustrated in Figure 6.2. The least mean square error point corresponds to the minimum error power. At this operating point the mean square error surface has zero gradient. From Equation (6.5), the gradient of the mean square error function with respect to the filter coefficient vector is given by

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} \mathcal{E}[e^2(m)] &= -2\mathcal{E}[x(m)\mathbf{y}(m)] + 2\mathbf{w}^T \mathcal{E}[\mathbf{y}(m)\mathbf{y}^T(m)] \\ &= -2\mathbf{r}_{yx} + 2\mathbf{w}^T \mathbf{R}_{yy}\end{aligned}\quad (6.7)$$

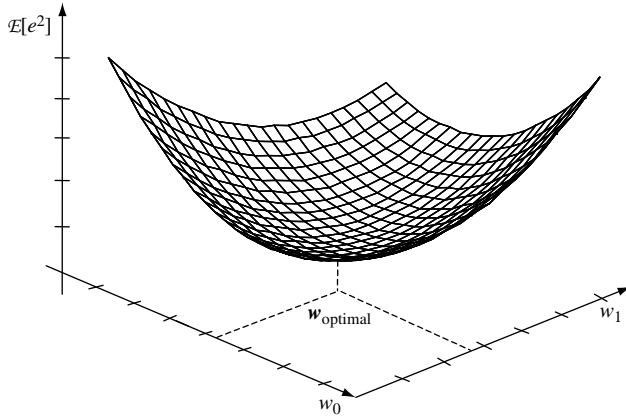


Figure 6.2 Mean square error surface for a two-tap FIR filter.

where the gradient vector is defined as

$$\frac{\partial}{\partial \mathbf{w}} = \left[\frac{\partial}{\partial w_0}, \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_{P-1}} \right]^T \tag{6.8}$$

The minimum mean square error Wiener filter is obtained by setting Equation (6.7) to zero:

$$\mathbf{R}_{yy} \mathbf{w} = \mathbf{r}_{yx} \tag{6.9}$$

or, equivalently,

$$\mathbf{w} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{yx} \tag{6.10}$$

In an expanded form, the Wiener filter solution, Equation (6.10), can be written as

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} = \begin{pmatrix} r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \dots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \dots & r_{yy}(P-2) \\ r_{yy}(2) & r_{yy}(1) & r_{yy}(0) & \dots & r_{yy}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & r_{yy}(P-3) & \dots & r_{yy}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{yx}(0) \\ r_{yx}(1) \\ r_{yx}(2) \\ \vdots \\ r_{yx}(P-1) \end{pmatrix} \tag{6.11}$$

From Equation (6.11), the calculation of the Wiener filter coefficients requires the autocorrelation matrix of the input signal and the cross-correlation vector of the input and the desired signals.

In statistical signal processing theory, the correlation values of a random process are obtained as the averages taken across the ensemble of different realisations of the process, as described in Chapter 3. However in many practical situations there are only one or two finite-duration realisations of the signals $x(m)$ and $y(m)$. Furthermore most signals are

nonstationary and need to be segmented in quasistationary short segments. In such cases, assuming the signals are correlation-ergodic, we can use time averages instead of ensemble averages. For a signal record of length N samples, the time-averaged correlation values are computed as

$$r_{yy}(k) = \frac{1}{N} \sum_{m=0}^{N-1} y(m)y(m+k) \quad (6.12)$$

Note from Equation (6.11) that the autocorrelation matrix \mathbf{R}_{yy} has a highly regular Toeplitz structure. A Toeplitz matrix has identical elements along the left–right diagonals of the matrix. Furthermore, the correlation matrix is also symmetric about the main diagonal elements. There are a number of efficient methods for solving the linear matrix Equation (6.11), including the Cholesky decomposition, the singular value decomposition and the QR decomposition (Section 6.2.1) methods.

6.2 BLOCK-DATA FORMULATION OF THE WIENER FILTER

In this section we consider an alternative formulation of a Wiener filter for a segment of N samples of the input signal $[y(0), y(1), \dots, y(N-1)]$ and the desired signal $[x(0), x(1), \dots, x(N-1)]$. The set of N linear equations describing the Wiener filter input–output relation can be written in matrix form as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = \begin{pmatrix} y(0) & y(-1) & y(-2) & \cdots & y(2-P) & y(1-P) \\ y(1) & y(0) & y(-1) & \cdots & y(3-P) & y(2-P) \\ y(2) & y(1) & y(0) & \cdots & y(4-P) & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(N-2) & y(N-3) & y(N-4) & \cdots & y(N-P) & y(N-1-P) \\ y(N-1) & y(N-2) & y(N-3) & \cdots & y(N+1-P) & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-2} \\ w_{P-1} \end{pmatrix} \quad (6.13)$$

Equation (6.13) can be rewritten in compact matrix notation as

$$\hat{\mathbf{x}} = \mathbf{Y}\mathbf{w} \quad (6.14)$$

The Wiener filter error is the difference between the desired signal and the filter output defined as

$$\begin{aligned} \mathbf{e} &= \mathbf{x} - \hat{\mathbf{x}} \\ &= \mathbf{x} - \mathbf{Y}\mathbf{w} \end{aligned} \quad (6.15)$$

The energy of the error vector, that is the sum of the squared elements of the error vector, is given by the inner vector product as

$$\begin{aligned} \mathbf{e}^T \mathbf{e} &= (\mathbf{x} - \mathbf{Y}\mathbf{w})^T (\mathbf{x} - \mathbf{Y}\mathbf{w}) \\ &= \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{Y}\mathbf{w} - \mathbf{w}^T \mathbf{Y}^T \mathbf{x} + \mathbf{w}^T \mathbf{Y}^T \mathbf{Y}\mathbf{w} \end{aligned} \quad (6.16)$$

The gradient of the squared error function with respect to the Wiener filter coefficients is obtained by differentiating Equation (6.16) w.r.t. \mathbf{w} as:

$$\frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{w}} = -2\mathbf{x}^T \mathbf{Y} + 2\mathbf{w}^T \mathbf{Y}^T \mathbf{Y} \quad (6.17)$$

The Wiener filter coefficients are obtained by setting the gradient of the squared error function of Equation (6.17) to zero; this yields

$$(\mathbf{Y}^T \mathbf{Y}) \mathbf{w} = \mathbf{Y}^T \mathbf{x} \quad (6.18)$$

or

$$\mathbf{w} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{x} \quad (6.19)$$

Note that the matrix $\mathbf{Y}^T \mathbf{Y}$ is a time-averaged estimate of the autocorrelation matrix of the filter input signal \mathbf{R}_{yy} , and that the vector $\mathbf{Y}^T \mathbf{x}$ is a time-averaged estimate of \mathbf{r}_{xy} , the cross-correlation vector of the input and the desired signals. Since the least square error method described in this section requires a block of N samples of the input and the desired signals, it is also referred to as the block least square (BLS) error estimation method. The block estimation method is appropriate for processing signals that can be considered as time-invariant over the duration of the block.

Theoretically, the Wiener filter is obtained from minimisation of the squared error across the ensemble of different realisations of a process, as described in the previous section. For a correlation-ergodic process, as the signal length N approaches infinity, the block-data Wiener filter of Equation (6.19) approaches the Wiener filter of Equation (6.10):

$$\lim_{N \rightarrow \infty} [\mathbf{w} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{x}] = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy} \quad (6.20)$$

6.2.1 QR DECOMPOSITION OF THE LEAST SQUARE ERROR EQUATION

An efficient and robust method for solving the least square error Equation (6.19) is the QR decomposition (QRD) method. In this method, the $N \times P$ signal matrix \mathbf{Y} [shown in Equation (6.13)] is decomposed into the product of an $N \times N$ orthonormal matrix \mathbf{Q} and a $P \times P$ upper-triangular matrix \mathcal{R} as

$$\mathbf{QY} = \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \quad (6.21)$$

where $\mathbf{0}$ is the $(N - P) \times P$ null matrix, \mathbf{Q} is an orthonormal matrix

$$\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I} \quad (6.22)$$

and the upper-triangular matrix \mathcal{R} is of the form

$$\mathcal{R} = \begin{pmatrix} r_{00} & r_{01} & r_{02} & r_{03} & \cdots & r_{0P-1} \\ 0 & r_{11} & r_{12} & r_{13} & \cdots & r_{1P-1} \\ 0 & 0 & r_{22} & r_{23} & \cdots & r_{2P-1} \\ 0 & 0 & 0 & r_{33} & \cdots & r_{3P-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_{P-1P-1} \end{pmatrix} \quad (6.23)$$

From Equations (6.21) and (6.22) we have

$$\mathbf{Y} = \mathbf{Q}^T \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \quad (6.24)$$

Substitution of Equation (6.24) in Equation (6.18) yields

$$\begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix}^T \mathbf{Q} \mathbf{Q}^T \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \mathbf{w} = \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix}^T \mathbf{Q} \mathbf{x} \quad (6.25)$$

From Equation (6.25) we have

$$\begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \mathbf{w} = \mathbf{Q} \mathbf{x} \quad (6.26)$$

From Equation (6.26) we have

$$\mathcal{R} \mathbf{w} = \mathbf{x}_Q \quad (6.27)$$

where the vector \mathbf{x}_Q on the right-hand side of Equation (6.27) is composed of the first P elements of the product $\mathbf{Q} \mathbf{x}$. Since the matrix \mathcal{R} is upper-triangular, the coefficients of the least square error filter can be obtained easily through a process of back substitution from Equation (6.27), starting with the coefficient $w_{P-1} = x_Q(P-1)/r_{P-1P-1}$.

The main computational steps in the QR decomposition are the determination of the orthonormal matrix, \mathbf{Q} , and the upper triangular matrix, \mathcal{R} . The decomposition of a matrix into \mathbf{QR} matrices can be achieved using a number of methods, including the Gram-Schmidt orthogonalisation method, the Householder method and the Givens rotation method.

6.3 INTERPRETATION OF WIENER FILTERS AS PROJECTIONS IN VECTOR SPACE

In this section, we consider an alternative formulation of Wiener filters where the least square error estimate is visualized as the perpendicular minimum distance *projection* of the desired signal vector onto the vector space of the input signal. A vector space is the collection of an infinite number of vectors that can be obtained from linear combinations of a number of independent vectors.

In order to develop a vector space interpretation of the least square error estimation problem, we rewrite the matrix Equation (6.11) and express the filter output signal vector $\hat{\mathbf{x}}$ as a linear weighted combination of the column vectors of the input signal matrix as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = w_0 \begin{pmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N-2) \\ y(N-1) \end{pmatrix} + w_1 \begin{pmatrix} y(-1) \\ y(0) \\ y(1) \\ \vdots \\ y(N-3) \\ y(N-2) \end{pmatrix} + \cdots + w_{P-1} \begin{pmatrix} y(1-P) \\ y(2-P) \\ y(3-P) \\ \vdots \\ y(N-1-P) \\ y(N-P) \end{pmatrix} \quad (6.28)$$

In compact notation, Equation (6.28) may be written as

$$\hat{\mathbf{x}} = w_0 \mathbf{y}_0 + w_1 \mathbf{y}_1 + \cdots + w_{P-1} \mathbf{y}_{P-1} \quad (6.29)$$

In Equation (6.29) the Wiener filter output $\hat{\mathbf{x}}$ is expressed as a linear combination of P basis vectors $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{P-1}]$, and hence it can be said that the estimate $\hat{\mathbf{x}}$ is in the vector subspace formed by the input signal vectors $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{P-1}]$.

In general, the N -dimensional input signal vectors $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{P-1}]$ in Equation (6.29) define the *basis* vectors for a subspace in an N -dimensional signal space. If the number of basis vectors P is equal to the vector dimension N , then the subspace encompasses the entire N -dimensional signal space and includes the desired signal vector \mathbf{x} . In this case, the signal estimate $\hat{\mathbf{x}} = \mathbf{x}$ and the estimation error is zero. However, in practice, $N > P$, and the signal space defined by the P input signal vectors of Equation (6.29) is only a subspace of the N -dimensional signal space. In this case, the estimation error is zero only if the desired signal \mathbf{x} happens to be in the subspace of the input signal, otherwise the best estimate of \mathbf{x} is the perpendicular projection of the vector \mathbf{x} onto the vector space of the input signal $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{P-1}]$, as explained in the following example.

Example 6.1

Figure 6.3 illustrates a vector space interpretation of a simple least square error estimation problem, where $\mathbf{y}^T = [y(m), y(m-1), y(m-2), y(m-3)]$ is the input observation signal, $\mathbf{x}^T = [x(m), x(m-1), x(m-2)]$ is the desired signal and $\mathbf{w}^T = [w_0, w_1]$ is the filter coefficient vector. As in Equation (6.26), the filter output can be written as

$$\begin{pmatrix} \hat{x}(m) \\ \hat{x}(m-1) \\ \hat{x}(m-2) \end{pmatrix} = w_0 \begin{pmatrix} y(m) \\ y(m-1) \\ y(m-2) \end{pmatrix} + w_1 \begin{pmatrix} y(m-1) \\ y(m-2) \\ y(m-3) \end{pmatrix} \quad (6.30)$$

In Equation (6.28), the filter input signal vectors $\mathbf{y}_1^T = [y(m), y(m-1), y(m-2)]$ and $\mathbf{y}_2^T = [y(m-1), y(m-2), y(m-3)]$ are three-dimensional vectors. The subspace defined by the linear combinations of the two input vectors $[\mathbf{y}_1, \mathbf{y}_2]$ is a two-dimensional plane in a three-dimensional space. The filter output is a linear combination of \mathbf{y}_1 and \mathbf{y}_2 , and hence it is confined to the plane containing these two vectors. The least square error estimate of \mathbf{x} is the orthogonal projection of \mathbf{x} on the plane of $[\mathbf{y}_1, \mathbf{y}_2]$, as shown by the shaded vector $\hat{\mathbf{x}}$. If the desired vector happens to be in the plane defined by the vectors \mathbf{y}_1 and \mathbf{y}_2 , then the estimation error will be zero, otherwise the estimation error will be the perpendicular distance of \mathbf{x} from the plane containing \mathbf{y}_1 and \mathbf{y}_2 .

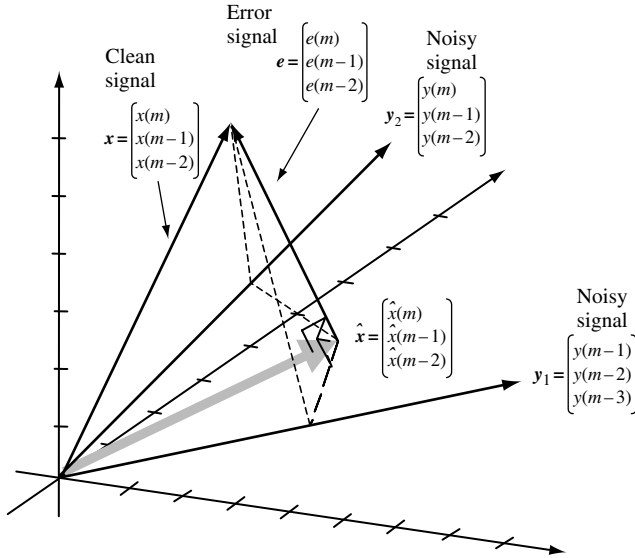


Figure 6.3 The least square error projection of a desired signal vector x onto a plane containing the input signal vectors y_1 and y_2 is the perpendicular projection of x , shown as the shaded vector.

6.4 ANALYSIS OF THE LEAST MEAN SQUARE ERROR SIGNAL

The optimality criterion in the formulation of the Wiener filter is the least mean square distance between the filter output and the desired signal. In this section, the variance of the filter error signal is analysed. Substituting the Wiener equation $R_{yy}w = r_{yx}$ in the mean squared error Equation (6.5) gives the least mean square error:

$$\begin{aligned} \mathcal{E}[e^2(m)] &= r_{xx}(0) - w^T r_{yx} \\ &= r_{xx}(0) - w^T R_{yy} w \end{aligned} \tag{6.31}$$

Now, for zero-mean signals, it is easy to show that the term $w^T R_{yy} w$ in Equation (6.31) is the variance of the Wiener filter output $\hat{x}(m)$:

$$\sigma_{\hat{x}}^2 = \mathcal{E}[\hat{x}^2(m)] = w^T R_{yy} w \tag{6.32}$$

Therefore Equation (6.31) may be written as

$$\sigma_e^2 = \sigma_x^2 - \sigma_{\hat{x}}^2 \tag{6.33}$$

where $\sigma_x^2 = \mathcal{E}[x^2(m)] = r_{xx}(0)$, $\sigma_{\hat{x}}^2 = \mathcal{E}[\hat{x}^2(m)]$ and $\sigma_e^2 = \mathcal{E}[e^2(m)]$ are the variances of the desired signal, the filter output, i.e. the estimate of the desired signal, and the error signal, respectively. In general, the filter input, $y(m)$, is composed of a signal component, $x_c(m)$, and a random noise, $n(m)$:

$$y(m) = x_c(m) + n(m) \tag{6.34}$$

where the signal $x_c(m)$ is that part of the observation $y(m)$ that is correlated with the desired signal $x(m)$, and it is this part of the input signal that may be transformable through a Wiener filter to the desired signal. Using Equation (6.34) the Wiener filter error may be decomposed into two distinct components:

$$\begin{aligned} e(m) &= x(m) - \sum_{k=0}^P w_k y(m-k) \\ &= \left[x(m) - \sum_{k=0}^P w_k x_c(m-k) \right] - \sum_{k=0}^P w_k n(m-k) \end{aligned} \quad (6.35)$$

or

$$e(m) = e_x(m) + e_n(m) \quad (6.36)$$

where $e_x(m)$ is the difference between the desired signal $x(m)$ and the output of the filter in response to the input signal component $x_c(m)$, i.e.

$$e_x(m) = x(m) - \sum_{k=0}^{P-1} w_k x_c(m-k) \quad (6.37)$$

and $e_n(m)$ is the error in the filter output due to the presence of noise $n(m)$ in the input signal:

$$e_n(m) = - \sum_{k=0}^{P-1} w_k n(m-k) \quad (6.38)$$

The variance of filter error can be rewritten as

$$\sigma_e^2 = \sigma_{e_x}^2 + \sigma_{e_n}^2 \quad (6.39)$$

Note that in Equation (6.36), $e_x(m)$ is that part of the signal that cannot be recovered by the Wiener filter, and represents part of the distortion in the filter output signal, and $e_n(m)$ is that part of the noise that cannot be blocked by the Wiener filter. Ideally, $e_x(m) = 0$ and $e_n(m) = 0$, but this ideal situation is possible only if the following conditions are satisfied:

- (1) the spectra of the signal and the noise are separable by a linear filter;
- (2) the signal component of the input, that is $x_c(m)$, is *linearly* transformable to $x(m)$;
- (3) the filter length, P , is sufficiently large. The issue of signal and noise separability is addressed in Section 6.6.

6.5 FORMULATION OF WIENER FILTERS IN THE FREQUENCY DOMAIN

In the frequency domain, the Wiener filter output, $\hat{X}(f)$, is the product of the input signal, $Y(f)$, and the filter frequency response, $W(f)$:

$$\hat{X}(f) = W(f)Y(f) \quad (6.40)$$

The estimation error signal $E(f)$ is defined as the difference between the desired signal $X(f)$ and the filter output $\hat{X}(f)$ as

$$\begin{aligned} E(f) &= X(f) - \hat{X}(f) \\ &= X(f) - W(f)Y(f) \end{aligned} \quad (6.41)$$

and the mean square error at a frequency f is given by

$$\mathcal{E} \left[|E(f)|^2 \right] = \mathcal{E} \left\{ [X(f) - W(f)Y(f)]^* [X(f) - W(f)Y(f)] \right\} \quad (6.42)$$

where $\mathcal{E}[\cdot]$ is the expectation function, and the symbol $*$ denotes the complex conjugate. Note from Parseval's theorem the mean square error in time and frequency domains are related by

$$\sum_{m=0}^{N-1} e^2(m) = \int_{-1/2}^{1/2} |E(f)|^2 df \quad (6.43)$$

To obtain the least mean square error filter we set the complex derivative of Equation (6.40) with respect to filter $W(f)$ to zero

$$\frac{\partial \mathcal{E}[|E(f)|^2]}{\partial W(f)} = 2W(f)P_{YY}(f) - 2P_{XY}(f) = 0 \quad (6.44)$$

where $P_{YY}(f) = \mathcal{E}[Y(f)Y^*(f)]$ and $P_{XY}(f) = \mathcal{E}[X(f)Y^*(f)]$ are the power spectrum of $Y(f)$, and the cross-power spectrum of $Y(f)$ and $X(f)$, respectively. From Equation (6.44), the least mean square error Wiener filter in the frequency domain is given as

$$W(f) = \frac{P_{XY}(f)}{P_{YY}(f)} \quad (6.45)$$

Alternatively, the frequency Wiener filter Equation (6.45) can be obtained from the Fourier transform of the time-domain Wiener Equation (6.9):

$$\sum_m \sum_{k=0}^{P-1} w_k r_{yy}(m-k) e^{-j\omega m} = \sum_m r_{yx}(m) e^{-j\omega m} \quad (6.46)$$

From the Wiener-Khinchine relation, correlation and power-spectral functions are Fourier transform pairs. Using this relation, and the Fourier transform property that convolution in time is equivalent to multiplication in frequency, Equation (6.46) can be transformed into frequency as

$$W(f)P_{YY}(f) = P_{XY}(f) \quad (6.47)$$

Re-arrangement of Equation (6.47) gives Equation (6.45).

6.6 SOME APPLICATIONS OF WIENER FILTERS

In this section, we consider some applications of the Wiener filter in reducing broadband additive noise, in time-alignment of signals in multichannel or multisensor systems, and in communication channel equalisation.

6.6.1 WIENER FILTERS FOR ADDITIVE NOISE REDUCTION

Consider a signal $x(m)$ observed in a broadband additive noise $n(m)$, and model as

$$y(m) = x(m) + n(m) \quad (6.48)$$

Assuming that the signal and the noise are uncorrelated, i.e. $r_{xn}(m) = 0$, it follows that the autocorrelation matrix of the noisy signal is the sum of the autocorrelation matrix of the signal $x(m)$ and the noise $n(m)$:

$$\mathbf{R}_{yy} = \mathbf{R}_{xx} + \mathbf{R}_{nn} \quad (6.49)$$

and we can also write

$$\mathbf{r}_{xy} = \mathbf{r}_{xx} \quad (6.50)$$

where \mathbf{R}_{yy} , \mathbf{R}_{xx} and \mathbf{R}_{nn} are the autocorrelation matrices of the noisy signal, the noise-free signal and the noise, respectively, and \mathbf{r}_{xy} is the cross-correlation vector of the noisy signal and the noise-free signal. Substitution of Equations (6.49) and (6.50) in the Wiener filter, Equation (6.10), yields

$$\mathbf{w} = (\mathbf{R}_{xx} + \mathbf{R}_{nn})^{-1} \mathbf{r}_{xx} \quad (6.51)$$

Equation (6.51) is the optimal linear filter for the removal of additive noise. In the following, a study of the frequency response of the Wiener filter provides useful insight into the operation of the Wiener filter. In the frequency domain, the noisy signal $Y(f)$ is given by

$$Y(f) = X(f) + N(f) \quad (6.52)$$

where $X(f)$ and $N(f)$ are the signal and noise spectra. For a signal observed in additive random noise, the frequency Wiener filter is obtained as

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \quad (6.53)$$

where $P_{XX}(f)$ and $P_{NN}(f)$ are the signal and noise power spectra. Dividing the numerator and the denominator of Equation (6.53) by the noise power spectra $P_{NN}(f)$ and substituting the variable $\text{SNR}(f) = P_{XX}(f)/P_{NN}(f)$ yields

$$W(f) = \frac{\text{SNR}(f)}{\text{SNR}(f) + 1} \quad (6.54)$$

where SNR is a signal-to-noise ratio measure. Note that the variable $\text{SNR}(f)$ is expressed in terms of the power-spectral ratio, and not in the more usual terms of log power ratio. Therefore $\text{SNR}(f) = 0$ corresponds to zero signal content or $-\infty$ dB.

From Equation (6.54), the following interpretation of the Wiener filter frequency response, $W(f)$, in terms of the signal-to-noise ratio can be deduced. For additive noise, the Wiener filter frequency response is a real positive number in the range $0 \leq W(f) \leq 1$. Now consider the two limiting cases of (a) a noise-free signal $\text{SNR}(f) = \infty$ and (b) an extremely noisy signal $\text{SNR}(f) = 0$. At very high SNR, $W(f) \approx 1$, and the filter applies little or no attenuation to the noise-free frequency component. At the other extreme, when $\text{SNR}(f) = 0$, $W(f) = 0$. Therefore, for additive noise, the Wiener filter attenuates each frequency component in proportion to an estimate of the signal to noise ratio. Figure 6.4 shows the variation of the Wiener filter response, $W(f)$, with the signal-to-noise ratio, $\text{SNR}(f)$.

An alternative illustration of the variations of the Wiener filter frequency response with $\text{SNR}(f)$ is shown in Figure 6.5. It illustrates the similarity between the Wiener filter frequency response and the signal spectrum for the case of an additive white noise disturbance. Note that, at a spectral peak of the signal spectrum, where the $\text{SNR}(f)$ is relatively high, the Wiener filter frequency response is also high, and the filter applies little attenuation. At a signal trough, the signal-to-noise ratio is low, and so is the Wiener filter response. Hence, for additive white noise, the Wiener filter response broadly follows the signal spectrum.

6.6.2 WIENER FILTERS AND SEPARABILITY OF SIGNAL AND NOISE

A signal is completely recoverable from noise if the spectra of the signal and the noise do not overlap. An example of a noisy signal with separable signal and noise spectra is shown in Figure 6.6(a). In this case, the signal and the noise occupy different parts of the frequency spectrum, and can be separated with a low-pass, or a high-pass, filter. Figure 6.6(b) illustrates a more common example of a signal and noise process with overlapping spectra. For this case, it is not possible to completely separate the signal from the noise. However,

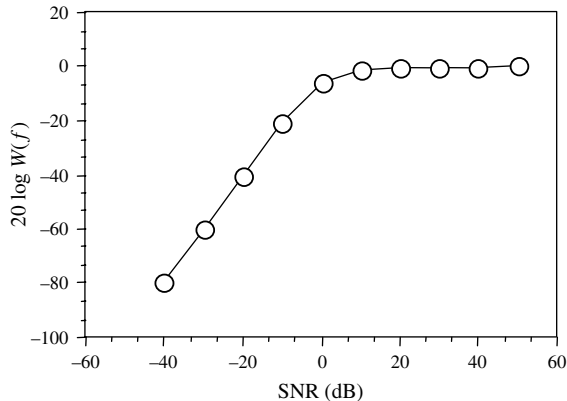


Figure 6.4 Variation of the gain of Wiener filter frequency response with SNR.

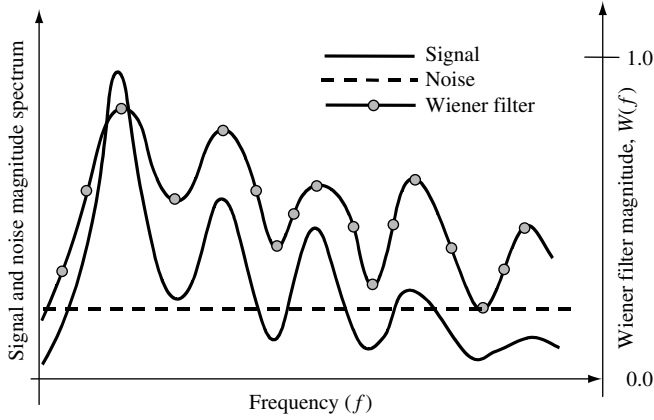


Figure 6.5 Illustration of the variation of Wiener frequency response with signal spectrum for additive white noise. The Wiener filter response broadly follows the signal spectrum.

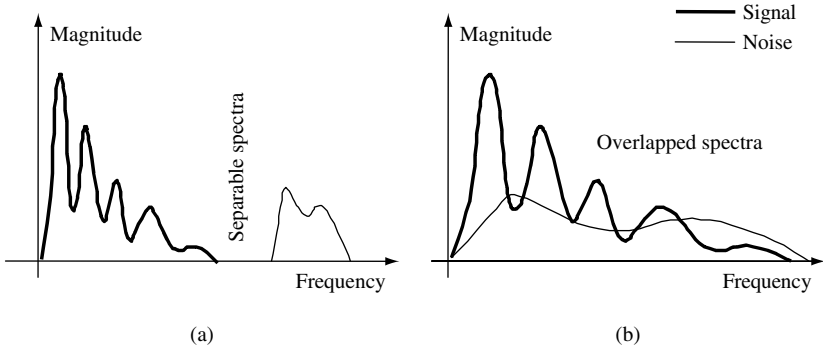


Figure 6.6 Illustration of separability: (a) the signal and noise spectra do not overlap, and the signal can be recovered by a low-pass filter; (b) the signal and noise spectra overlap, and the noise can be reduced but not completely removed.

the effects of the noise can be reduced by using a Wiener filter that attenuates each noisy signal frequency in proportion to an estimate of the signal-to-noise ratio, as described by Equation (6.54).

6.6.3 THE SQUARE-ROOT WIENER FILTER

In the frequency domain, the Wiener filter output, $\hat{X}(f)$, is the product of the input frequency, $Y(f)$, and the filter response, $W(f)$, as expressed in Equation (6.40). Taking the expectation of the squared magnitude of both sides of Equation (6.40) yields the power spectrum of the

filtered signal as

$$\begin{aligned}\mathcal{E}[|\hat{X}(f)|^2] &= |W(f)|^2 \mathcal{E}[|Y(f)|^2] \\ &= |W(f)|^2 P_{YY}(f)\end{aligned}\quad (6.55)$$

Substitution of $W(f)$ from Equation (6.45) in Equation (6.55) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XY}^2(f)}{P_{YY}(f)} \quad (6.56)$$

Now, for a signal observed in an uncorrelated additive noise, we have

$$P_{YY}(f) = P_{XX}(f) + P_{NN}(f) \quad (6.57)$$

and

$$P_{XY}(f) = P_{XX}(f) \quad (6.58)$$

Substitution of Equations (6.57) and (6.58) in Equation (6.56) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XX}^2(f)}{P_{XX}(f) + P_{NN}(f)} \quad (6.59)$$

Now, in Equation (6.40) if instead of the Wiener filter, the square root of the Wiener filter magnitude frequency response is used, the result is

$$\hat{X}(f) = |W(f)|^{1/2} Y(f) \quad (6.60)$$

and the power spectrum of the signal, filtered by the square-root Wiener filter, is given by

$$\mathcal{E}[|\hat{X}(f)|^2] = \left[|W(f)|^{1/2} \right]^2 \mathcal{E}[|Y(f)|^2] = \frac{P_{XY}(f)}{P_{YY}(f)} P_{YY}(f) = P_{XY}(f) \quad (6.61)$$

Now, for uncorrelated signal and noise Equation (6.58) becomes

$$\mathcal{E}[|\hat{X}(f)|^2] = P_{XX}(f) \quad (6.62)$$

Thus, for additive noise the power spectrum of the output of the square-root Wiener filter is the same as the power spectrum of the desired signal.

6.6.4 WIENER CHANNEL EQUALISER

The distortions in a communication channel may be modelled by a combination of a linear filter and an additive random noise source, as shown in Figure 6.7. The input–output signals of a linear time-invariant channel can be modelled as

$$y(m) = \sum_{k=0}^{P-1} h_k x(m-k) + n(m) \quad (6.63)$$

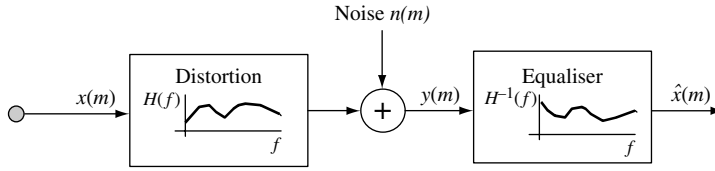


Figure 6.7 Illustration of a channel model followed by an equaliser.

where $x(m)$ and $y(m)$ are the transmitted and received signals, $[h_k]$ is the impulse response of a linear filter model of the channel, and $n(m)$ models the channel noise. In the frequency domain Equation (6.63) becomes

$$Y(f) = X(f)H(f) + N(f) \quad (6.64)$$

where $X(f)$, $Y(f)$, $H(f)$ and $N(f)$ are the signal, noisy signal, channel and noise spectra, respectively. To remove the channel distortions, the receiver is followed by an equaliser. The equaliser's input is the distorted signal at the channel output, and the desired signal is the clean signal at the channel input. Using Equation (6.45), it is easy to show that the frequency domain Wiener equaliser is given by

$$W(f) = \frac{P_{XX}(f)H^*(f)}{P_{XX}(f)|H(f)|^2 + P_{NN}(f)} \quad (6.65)$$

where it is assumed that the signal and the channel noise are uncorrelated. In the absence of channel noise, $P_{NN}(f) = 0$, and the Wiener filter is simply the inverse of the channel distortion model $W(f) = H^{-1}(f)$. The equalisation problem is treated in detail in Chapter 15.

6.6.5 TIME-ALIGNMENT OF SIGNALS IN MULTICHANNEL/MULTISENSOR SYSTEMS

In multichannel/multisensor signal processing there is an array of noisy and distorted versions of a signal $x(m)$, and the objective is to use all the observations in estimating $x(m)$, as illustrated in Figure 6.8, where the phase and frequency characteristics of each channel are modelled by a linear filter $h(m)$.

As a simple example, consider the problem of time-alignment of two noisy records of a signal given as

$$y_1(m) = x(m) + n_1(m) \quad (6.66)$$

$$y_2(m) = Ax(m - D) + n_2(m) \quad (6.67)$$

where $y_1(m)$ and $y_2(m)$ are the noisy observations from channels 1 and 2, $n_1(m)$ and $n_2(m)$ are uncorrelated noise in each channel, D is the relative time delay of arrival of the two signals, and A is an amplitude scaling factor. Now assume that $y_1(m)$ is used as the input

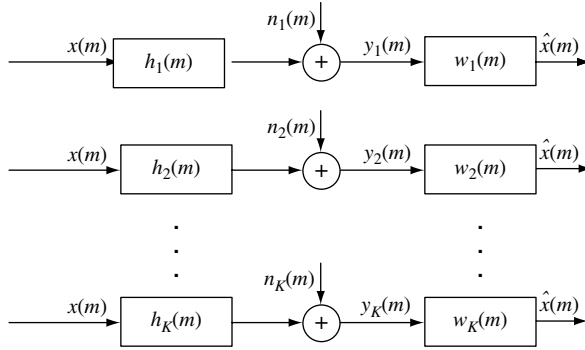


Figure 6.8 Illustration of a multichannel system where Wiener filters are used to time-align the signals from different channels.

to a Wiener filter and that, in the absence of the signal $x(m)$, $y_2(m)$ is used as the ‘desired’ signal. The error signal is given by

$$\begin{aligned}
 e(m) &= y_2(m) - \sum_{k=0}^{P-1} w_k y_1(m) \\
 &= \left[Ax(m-D) - \sum_{k=0}^{P-1} w_k x(m) \right] + \left[\sum_{k=0}^{P-1} w_k n_1(m) \right] + n_2(m)
 \end{aligned} \tag{6.68}$$

The Wiener filter strives to minimise the terms shown inside the square brackets in Equation (6.68). Using the Wiener filter Equation (6.10), we have

$$\begin{aligned}
 \mathbf{w} &= \mathbf{R}_{y_1 y_1}^{-1} \mathbf{r}_{y_1 y_2} \\
 &= (\mathbf{R}_{xx} + \mathbf{R}_{n_1 n_1})^{-1} A \mathbf{r}_{xx}(D)
 \end{aligned} \tag{6.69}$$

where $\mathbf{r}_{xx}(D) = \mathcal{E}[x(m-D)x(m)]$. The frequency-domain equivalent of Equation (6.69) can be derived as

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{N_1 N_1}(f)} A e^{-j\omega D} \tag{6.70}$$

Note that in the absence of noise, the Wiener filter becomes a pure phase (or a pure delay) filter, $W(f) = A e^{-j\omega D}$, with a flat magnitude response.

6.7 IMPLEMENTATION OF WIENER FILTERS

The implementation of a Wiener filter for additive noise reduction, using Equations (6.51) or (6.53), requires the autocorrelation functions, or equivalently the power spectra, of the signal and noise. In speech recognition the power spectra, or autocorrelation functions of signal and noise, can be obtained from speech and noise models (see Chapters 5 and 16).

When statistical models of speech and noise are not available, the noise power spectrum can be obtained from the signal-inactive, noise-only, periods. The assumption is that the noise is quasistationary, and that its power spectra remain relatively stationary between the update periods. This is a reasonable assumption for many noisy environments, such as the noise inside a car emanating from the engine and wind, aircraft noise or office noise from computer machines.

The main practical problem in the implementation of a Wiener filter is that the desired signal is often observed in noise, and that the autocorrelation or power spectra of the desired signal are not readily available. Figure 6.9 illustrates the block-diagram configuration of a system for implementation of a Wiener filter for additive noise reduction. The implementation of this filter requires estimates of the spectral signal-to-noise ratio, $SNR(f)$.

The estimate of spectral signal-to-noise ratio is obtained from the estimates of the power spectra of the signal and noise. The noise estimate is obtained from speech inactive periods. An estimate of the clean signal power spectra may be obtained by subtracting an estimate of the noise spectra from that of the noisy signal.

A filter bank implementation of the Wiener filter is shown in Figure 6.10, where the incoming signal is divided into N sub-bands. A first-order integrator, placed at the output of each band-pass filter, gives an estimate of the power spectra of the noisy signal. The power spectrum of the original signal is obtained by subtracting an estimate of the noise power spectrum from the noisy signal.

In a Bayesian implementation of the Wiener filter, prior models of speech and noise, such as hidden Markov models, are used to obtain the power spectra of speech and noise required for calculation of the filter coefficients.

6.7.1 THE CHOICE OF WIENER FILTER ORDER

The choice of Wiener filter order affects:

- (1) the ability of the filter to model and remove distortions and reduce the noise;
- (2) the computational complexity of the filter;
- (3) the numerical stability of the of the Wiener solution – a large filter may produce an ill-conditioned large-dimensional correlation matrix in Equation (6.10).

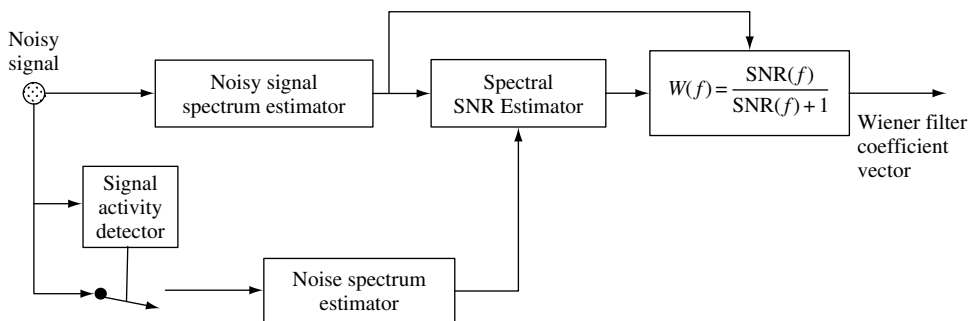


Figure 6.9 Configuration of a system for estimation of frequency Wiener filter.

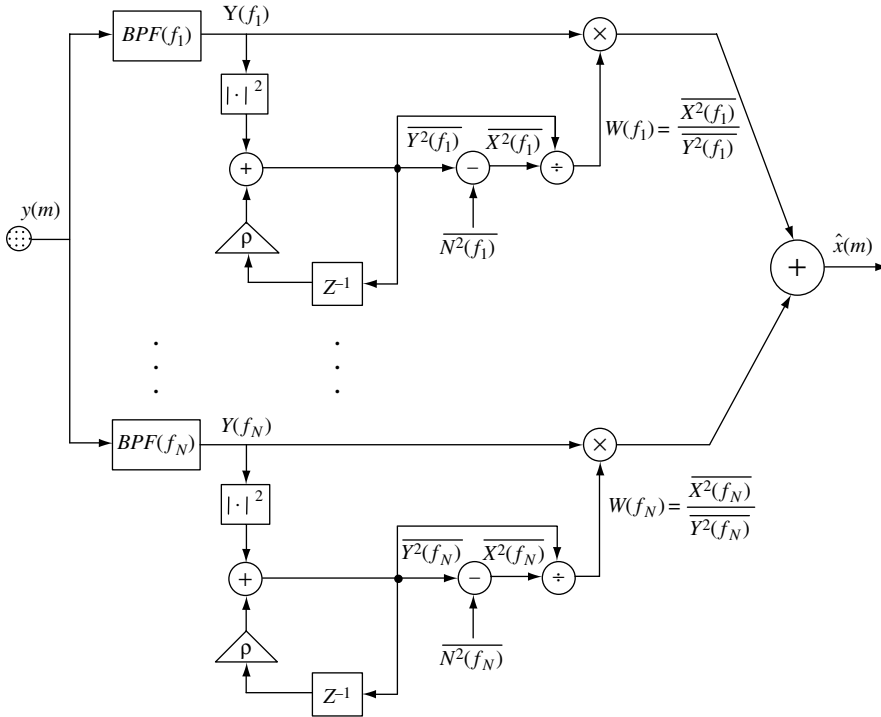


Figure 6.10 A filter-bank implementation of a Wiener filter for additive noise reduction.

The choice of the filter length also depends on the application and the method of implementation of the Wiener filter. For example, in a filter-bank implementation of the Wiener filter for additive noise reduction, Figure 6.10, the number of filter coefficients is equal to the number of filter banks, and typically the number of filter banks is between 16 to 64. On the other hand, for many applications a direct implementation of the time-domain Wiener filter requires a larger filter length, say between 64 and 256 taps.

A reduction in the required length of a time-domain Wiener filter can be achieved by dividing the time domain signal into N sub-band signals. Each sub-band signal can then be down-sampled by a factor of N . The down-sampling results in a reduction, by a factor of N , in the required length of each sub-band Wiener filter. In Chapter 14, a sub-band echo canceller is described.

6.7.2 IMPROVEMENTS TO WIENER FILTERS

The performance of Wiener filter can be limited by the following factors:

- (1) the signal-to-noise ratio – generally the Wiener filter performance deteriorates with decreasing SNR;
- (2) the signal nonstationarity – the Wiener filter theory assumes that the signal processes are stationary and any deviations from the assumption of stationarity will affect the ability

- of the filter to estimate and track the correlation or power spectrum functions needed for computation of the filter coefficients;
- (3) the Wiener filter is a linear filter and the presence of significant nonlinear distortion in the input will affect the filter performance.

The performance of the Wiener filter can be improved by the use of a spectral-time tracking and smoothing process employed to track and smooth the variations of the spectral components of the signals over time. For example, in noisy speech processing, the evolution over time of the significant spectral components of the signal and noise may be tracked in order to remove the fluctuations and errors in estimation of the correlation or spectral functions needed to compute Wiener filter coefficients.

6.8 SUMMARY

A Wiener filter is formulated to transform an input signal to an output that is as close to a desired signal as possible. This chapter began with the derivation of the least square error Wiener filter. In Section 6.2, we derived the block-data least square error Wiener filter for applications where only finite-length realisations of the input and the desired signals are available. In such cases, the filter is obtained by minimising a time-averaged squared error function. In Section 6.3, we considered a vector space interpretation of the Wiener filters as the perpendicular projection of the desired signal onto the space of the input signal.

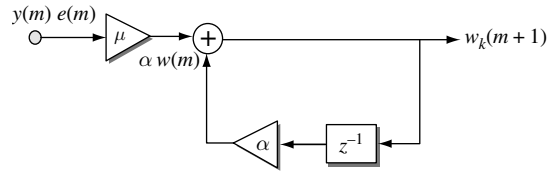
In Section 6.4, the least mean square error signal was analysed. The mean square error is zero only if the input signal is related to the desired signal through a linear and invertible filter. For most cases, owing to noise and/or nonlinear distortions of the input signal, the minimum mean square error would be nonzero. In Section 6.5, we derived the Wiener filter in the frequency domain, and considered the issue of separability of signal and noise using a linear filter. Finally, in Section 6.6, we considered some applications of Wiener filters in noise reduction, time-delay estimation and channel equalisation.

BIBLIOGRAPHY

- Akaike H. (1974) A new look at statistical model identification. *IEEE Trans. Autom. Control*, **AC-19**: 716–723.
- Alexander S.T. (1986) *Adaptive Signal Processing Theory and Applications*. Springer, New York.
- Anderson B.D. and Moor J.B. (1979) *Linear Optimal Control*. Prentice-Hall, Englewood Cliffs, NJ.
- Dorny C.N. (1975) *A Vector Space Approach to Models and Optimisation*. Wiley, New York.
- Durbin J. (1959) Efficient estimation of parameters in moving average models. *Biometrika*, **46**: 306–316.
- Giordano A.A. and Hsu F.M. (1985) *Least Square Estimation with Applications to Digital Signal Processing*. Wiley, New York.
- Givens W. (1958) Computation of plane unitary rotations transforming a general matrix to triangular form. *SIAM J. Appl. Math.*, **6**: 26–50.
- Golub G.H. and Reinsch (1970) Singular value decomposition and least squares solutions. *Numer. Math.*, **14**: 403–420.
- Golub G.H., and Van Loan C.F. (1980). An analysis of the total least squares problem. *SIAM J. Numer. Anal.*, **17**: 883–893.
- Golub G.H. and Van Loan C.F. (1983). *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD.
- Halmos P.R. (1974) *Finite-dimensional Vector Spaces*. Springer New York.
- Haykin S. (1991) *Adaptive Filter Theory*, 2nd edn. Prentice-Hall, Englewood Cliffs, NJ.
- Householder A.S. (1964) *The Theory of Matrices in Numerical Analysis*. Blaisdell, Waltham, MA.

- Kailath T. (1974) A view of three decades of linear filtering theory. *IEEE Trans. Inform. Theory*, **IT-20**: 146–181.
- Kailath T. (1977) *Linear Least Squares Estimation, Benchmark Papers in Electrical Engineering and Computer Science*. Dowden, Hutchinson & Ross, Stroudsburg, PA.
- Kailath T. (1980) *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- Klema V.C. and Laub A. J. (1980) The singular value decomposition: its computation and some applications. *IEEE Trans. Autom. Control*, **AC-25**: 164–176.
- Kolmogorov A.N. (1939) Sur l'interpolation et extrapolation des suites stationnaires. *C. R. Acad. Sci.*, **208**: 2043–2046.
- Lawson C.L. and Hanson R.J. (1974) *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ.
- Orfanidis S.J. (1988) *Optimum Signal Processing: an Introduction*, 2nd edn. Macmillan, New York.
- Scharf L.L. (1991) *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison Wesley, Reading, MA.
- Strang G. (1976) *Linear Algebra and its Applications*, 3rd edn. Harcourt Brace Jovanovich, San Diego, CA.
- Wiener N. (1949) *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. MIT Press, Cambridge, MA.
- Wilkinson J.H. (1965) *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford.
- Whittle P.W. (1983) *Prediction and Regulation by Linear Least-Squares Methods*. University of Minnesota Press, Minneapolis, MN.
- Wold H. (1954) *The Analysis of Stationary Time Series*, 2nd edn. Almqvist and Wicksell, Uppsala.

7



Adaptive Filters

7.1 Introduction	188
7.2 State-space Kalman Filters	188
7.3 Sample-adaptive Filters	195
7.4 Recursive Least Square Adaptive Filters	196
7.5 The Steepest-descent Method	201
7.6 The LMS Filter	204
7.7 Summary	207
Bibliography	208

Adaptive filters work on the principle of minimizing the mean squared difference (or error) between the filter output and a target (or desired) signal. Adaptive filters are used for estimation of nonstationary signals and systems, or in applications where a sample-by-sample adaptation of a process and/or a low processing delay is required. Applications of adaptive filters include multichannel noise reduction, radar/sonar signal processing, channel equalisation for cellular mobile phones, echo cancellation and low-delay speech coding.

This chapter begins with a study of the theory of state-space Kalman filter. In Kalman filter theory a state equation models the dynamics of the signal generation process, and an observation equation models the channel distortion and additive noise.

We study recursive least square (RLS) error adaptive filters. The RLS filter is a sample-adaptive formulation of the Wiener filter, and for stationary signals should converge to the same solution as the Wiener filter. In least square error filtering, an alternative to using a Wiener-type closed-form solution is an iterative gradient-based search for the optimal filter coefficients. The steepest-descent search is a gradient-based method for searching the least square error performance curve for the minimum error filter coefficients. We study the steepest-descent method, and then consider the computationally inexpensive LMS gradient search method.

7.1 INTRODUCTION

Adaptive filters are used in applications that involve a combination of three broad signal processing problems:

- (1) de-noising and channel equalization – filtering a time-varying noisy signal to remove the effect of noise and channel distortions;
- (2) trajectory estimation – tracking and prediction of the trajectory of a nonstationary signal or parameter observed in noise;
- (3) system identification – adaptive estimation of the parameters of a time-varying system from a related observation.

Adaptive linear filters work on the principle that the desired signal or parameters can be extracted from the input through a filtering or estimation operation. The adaptation of the filter parameters is based on minimizing the mean squared error between the filter output and a target (or desired) signal. The use of the LSE criterion is equivalent to the principal of orthogonality in which at any discrete time m the estimator is expected to use all the available information such that any estimation error at time m is orthogonal to all the information available up to time m .

An adaptive filter can be a combination of the following types of filters:

- single-input or multi-input filters;
- linear or nonlinear filters;
- finite impulse response FIR or infinite impulse response IIR filters.

In this chapter we are mainly concerned with linear FIR filters which, because of their stability and relative ease of adaptation, are the most widely used type of adaptive filter. The adaptation algorithm can be based on a variant of one of the three most commonly used adaptive estimation methods, namely

- Kalman filters;
- RLS filters;
- LMS filters.

The different types of adaptation algorithms differ mainly in terms of the prior or estimated knowledge of system function and covariance matrices of signal and noise and also in terms of the complexity of the solutions.

7.2 STATE-SPACE KALMAN FILTERS

Kalman filter, as illustrated in Figure 7.1, is a recursive least square error method for estimation of a signal distorted in transmission through a channel and observed in noise. Kalman filters can be used with time-varying as well as time-invariant processes.

Kalman filter theory is based on a state-space approach in which a state equation models the dynamics of the signal generation process and an observation equation models the noisy

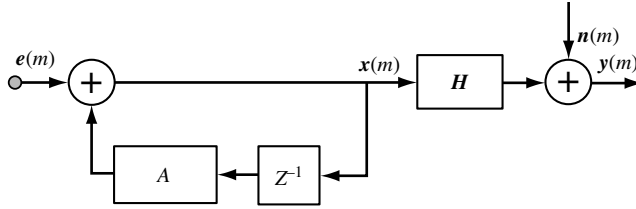


Figure 7.1 Illustration of signal and observation models in Kalman filter theory.

and distorted observation signal. For a signal $\mathbf{x}(m)$ and noisy observation $\mathbf{y}(m)$, equations describing the state process model and the observation model are defined as

$$\mathbf{x}(m) = A\mathbf{x}(m - 1) + \mathbf{e}(m) \tag{7.1}$$

$$\mathbf{y}(m) = H\mathbf{x}(m) + \mathbf{n}(m) \tag{7.2}$$

where $\mathbf{x}(m)$ is the P -dimensional signal, or the state parameter, vector at time m ; A is a $P \times P$ dimensional state transition matrix that relates the states of the process at times $m - 1$ and m ; $\mathbf{e}(m)$ is the P -dimensional uncorrelated input excitation vector of the state equation – $\mathbf{e}(m)$ is a normal (Gaussian) process, $p[\mathbf{e}(m)] \sim N(0, \mathbf{Q})$; \mathbf{Q} is the $P \times P$ covariance matrix of $\mathbf{e}(m)$; $\mathbf{y}(m)$ is the M -dimensional noisy and distorted observation vector; H is the $M \times P$ dimensional channel distortion matrix; $\mathbf{n}(m)$ is an M -dimensional noise vector, also known as measurement noise – $\mathbf{n}(m)$ is a normal (Gaussian) process, $p[\mathbf{n}(m)] \sim N(0, \mathbf{R})$; and \mathbf{R} is the $M \times M$ dimensional covariance matrix of $\mathbf{n}(m)$.

Kalman Filter Algorithm

Input: observation vectors $\{\mathbf{y}(m)\}$

Output: state or signal vectors $\{\hat{\mathbf{x}}(m)\}$

Initial conditions:

prediction error covariance matrix

$$\mathbf{P}(0|-1) = \delta \mathbf{I} \tag{7.3}$$

prediction

$$\hat{\mathbf{x}}(0|-1) = 0 \tag{7.4}$$

for $m = 0, 1, \dots$

Time-Update (Prediction) Equations

State prediction equation:

$$\hat{\mathbf{x}}(m|m - 1) = A\hat{\mathbf{x}}(m - 1) \tag{7.5}$$

(Continued)

Covariance matrix of prediction error:

$$\mathbf{P}(m|m-1) = \mathbf{A}\mathbf{P}(m-1)\mathbf{A}^T + \mathbf{Q} \quad (7.6)$$

Measurement-update (estimate) equations

Kalman gain vector:

$$\mathbf{K}(m) = \mathbf{P}(m|m-1)\mathbf{H}^T [\mathbf{H}\mathbf{P}(m|m-1)\mathbf{H}^T + \mathbf{R}]^{-1} \quad (7.7)$$

State update:

$$\hat{\mathbf{x}}(m) = \hat{\mathbf{x}}(m|m-1) + \mathbf{K}(m)[\mathbf{y}(m) - \mathbf{H}\hat{\mathbf{x}}(m|m-1)] \quad (7.8)$$

Covariance matrix of estimation error:

$$\mathbf{P}(m) = [\mathbf{I} - \mathbf{K}\mathbf{H}]\mathbf{P}(m|m-1) \quad (7.9)$$

7.2.1 DERIVATION OF THE KALMAN FILTER ALGORITHM

The Kalman filter can be derived as a recursive minimum mean square error estimator of a signal, $\mathbf{x}(m)$, from a noisy observation, $\mathbf{y}(m)$. The derivation of Kalman filter assumes that the state transition matrix, \mathbf{A} , the channel distortion matrix, \mathbf{H} , the covariance matrix, \mathbf{Q} , of the input $\mathbf{e}(m)$ of the state equation and the covariance matrix, \mathbf{R} , of the additive noise $\mathbf{n}(m)$ are given.

The derivation of Kalman filter, described next, is based on the following methodology:

- (1) *prediction (update) step* – the signal state is predicted from the previous observations and a prediction error covariance matrix is obtained, Equations (7.5)–(7.6);
- (2) *estimation (measurement) step* – the prediction from step 1 and the innovation (innovation is the difference between the prediction and noise observations) are used to make an estimate of the signal. At this stage the Kalman gain vector and estimation error covariance matrix are calculated, Equations (7.7)–(7.9).

In this chapter, we use the notation $\hat{\mathbf{y}}(m|m-i)$ to denote a prediction of $\mathbf{y}(m)$ based on the observation samples up to the time $m-i$. Assume that $\hat{\mathbf{x}}(m|m-1)$ is the least square error prediction of $\mathbf{x}(m)$ based on the observations $[\mathbf{y}(0), \dots, \mathbf{y}(m-1)]$. Define a prediction equation as

$$\hat{\mathbf{x}}(m|m-1) = \mathbf{A}\hat{\mathbf{x}}(m-1) \quad (7.10)$$

An *innovation* signal composed of prediction error plus noise may be defined as

$$\mathbf{v}(m) = \mathbf{y}(m) - \mathbf{H}\hat{\mathbf{x}}(m|m-1) \quad (7.11)$$

where $\hat{\mathbf{x}}(m|m-1)$ denotes the least square error prediction of the signal $\mathbf{x}(m)$. The innovation signal vector, $\mathbf{v}(m)$, is the part of $\mathbf{y}(m)$ that is unpredictable from past observations; it includes both the noise and the unpredictable part of the signal $\mathbf{x}(m)$.

For an optimal linear least mean square error estimate, the innovation signal must be an uncorrelated process orthogonal to the past observation vectors; hence we have

$$\mathcal{E}[\mathbf{v}(m)\mathbf{y}^T(m-k)] = 0 \quad k > 0 \quad (7.12)$$

and

$$\mathcal{E}[\mathbf{v}(m)\mathbf{v}^T(k)] = 0 \quad m \neq k \quad (7.13)$$

The concept of innovations is central to the derivation of the Kalman filter. The least square error criterion is satisfied if the estimation error is *orthogonal* to the past samples.

In the following derivation of the Kalman filter, the orthogonality condition of Equation (7.12) is used as the starting point to derive an optimal linear filter whose innovations $\mathbf{v}(m)$ are orthogonal to the past observations $\mathbf{y}(m)$.

Substituting the observation Equation (7.2) in Equation (7.11) and using the relationship

$$\begin{aligned} \hat{\mathbf{y}}(m|m-1) &= \mathcal{E}[\mathbf{y}(m)|\hat{\mathbf{x}}(m|m-1)] \\ &= \mathbf{H}\hat{\mathbf{x}}(m|m-1) \end{aligned} \quad (7.14)$$

yields

$$\begin{aligned} \mathbf{v}(m) &= \mathbf{H}\mathbf{x}(m) + \mathbf{n}(m) - \mathbf{H}\hat{\mathbf{x}}(m|m-1) \\ &= \mathbf{H}\tilde{\mathbf{x}}(m|m-1) + \mathbf{n}(m) \end{aligned} \quad (7.15)$$

where $\tilde{\mathbf{x}}(m|m-1)$ is the signal prediction error vector defined as

$$\tilde{\mathbf{x}}(m|m-1) = \mathbf{x}(m) - \hat{\mathbf{x}}(m|m-1) \quad (7.16)$$

From Equation (7.15) the covariance matrix of the innovation signal $\mathbf{v}(m)$ is given by

$$\mathcal{E}[\mathbf{v}(m)\mathbf{v}^T(m)] = \mathbf{H}\mathbf{P}(m|m-1)\mathbf{H}^T + \mathbf{R} \quad (7.17)$$

where $\mathbf{P}(m|m-1)$ is the covariance matrix of the prediction error $\tilde{\mathbf{x}}(m|m-1)$. The estimation of $\mathbf{x}(m)$, based on the samples available up to time m , can be expressed recursively as a linear combination of the prediction of $\mathbf{x}(m)$ based on the samples available up to time $m-1$, and the innovation signal at time m as

$$\hat{\mathbf{x}}(m) = \hat{\mathbf{x}}(m|m-1) + \mathbf{K}(m)\mathbf{v}(m) \quad (7.18)$$

where the $P \times M$ matrix $\mathbf{K}(m)$ is the Kalman gain matrix. Now, from Equation (7.1), we have

$$\hat{\mathbf{x}}(m|m-1) = \mathbf{A}\hat{\mathbf{x}}(m-1) \quad (7.19)$$

Substitution of Equation (7.19) in Equation (7.18) gives

$$\hat{\mathbf{x}}(m) = \mathbf{A}\hat{\mathbf{x}}(m-1) + \mathbf{K}(m)\mathbf{v}(m) \quad (7.20)$$

To obtain a recursive relation for the computation and update of the Kalman gain matrix, multiply both sides of Equation (7.18) by $\mathbf{v}^T(m)$ and take the expectation of the results to yield

$$\mathcal{E}[\hat{\mathbf{x}}(m)\mathbf{v}^T(m)] = \mathcal{E}[\hat{\mathbf{x}}(m|m-1)\mathbf{v}^T(m)] + \mathbf{K}(m)\mathcal{E}[\mathbf{v}(m)\mathbf{v}^T(m)] \quad (7.21)$$

Owing to the required orthogonality of the innovation sequence $\mathbf{v}(m)$ to the past samples, we have

$$\mathcal{E}[\hat{\mathbf{x}}(m|m-1)\mathbf{v}^T(m)] = 0 \quad (7.22)$$

Hence, from Equations (7.21) and (7.22), the Kalman gain matrix is given by

$$\mathbf{K}(m) = \mathcal{E}[\hat{\mathbf{x}}(m)\mathbf{v}^T(m)] \mathcal{E}[\mathbf{v}(m)\mathbf{v}^T(m)]^{-1} \quad (7.23)$$

The first term on the right-hand side of Equation (7.23) can be expressed as

$$\begin{aligned} \mathcal{E}[\hat{\mathbf{x}}(m)\mathbf{v}^T(m)] &= \mathcal{E}[[\mathbf{x}(m) - \tilde{\mathbf{x}}(m)]\mathbf{v}^T(m)] \\ &= \mathcal{E}[[\hat{\mathbf{x}}(m|m-1) + \tilde{\mathbf{x}}(m|m-1) - \tilde{\mathbf{x}}(m)]\mathbf{v}^T(m)] \\ &= \mathcal{E}[\hat{\mathbf{x}}(m|m-1)\mathbf{v}^T(m)] \\ &= \mathcal{E}[\tilde{\mathbf{x}}(m|m-1)[\mathbf{H}\mathbf{x}(m) + \mathbf{n}(m) - \mathbf{H}\hat{\mathbf{x}}(m|m-1)]^T] \\ &= \mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\tilde{\mathbf{x}}^T(m|m-1)]\mathbf{H}^T \\ &= \mathbf{P}(m|m-1)\mathbf{H}^T \end{aligned} \quad (7.24)$$

In developing the successive lines of Equation (7.24), we have used the following relations:

$$\mathbf{x}(m) = \hat{\mathbf{x}}(m|m-1) + \tilde{\mathbf{x}}(m|m-1) \quad (7.25)$$

$$\mathcal{E}[\tilde{\mathbf{x}}(m)\mathbf{v}^T(m)] = 0 \quad (7.26)$$

$$\mathcal{E}[\hat{\mathbf{x}}(m|m-1)\mathbf{v}^T(m)] = 0 \quad (7.27)$$

$$\mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\mathbf{n}^T(m)] = 0 \quad (7.28)$$

Substitution of Equations (7.17) and (7.24) in Equation (7.23) yields the following equation for the Kalman gain matrix:

$$\mathbf{K}(m) = \mathbf{P}(m|m-1)\mathbf{H}^T [\mathbf{H}\mathbf{P}(m|m-1)\mathbf{H}^T + \mathbf{R}]^{-1} \quad (7.29)$$

where $\mathbf{P}(m|m-1)$ is the covariance matrix of the signal prediction error $\tilde{\mathbf{x}}(m|m-1)$. Note that the Kalman gain vector can be interpreted as a function of signal-to-noise ratio of the innovation signal.

A recursive relation for calculation of $\mathbf{P}(m|m-1)$, the covariance matrix of prediction error $\tilde{\mathbf{x}}(m|m-1)$ is derived as follows

$$\tilde{\mathbf{x}}(m|m-1) = \mathbf{x}(m) - \hat{\mathbf{x}}(m|m-1) \quad (7.30)$$

Substitution of Equation (7.1) and (7.19) in Equation (7.30) and rearrangement of the terms yields

$$\begin{aligned} \tilde{\mathbf{x}}(m|m-1) &= [\mathbf{A}\mathbf{x}(m-1) + \mathbf{e}(m)] - [\mathbf{A}\hat{\mathbf{x}}(m-1)] \\ &= \mathbf{A}\tilde{\mathbf{x}}(m-1) + \mathbf{e}(m) \end{aligned} \quad (7.31)$$

The covariance matrix of $\tilde{\mathbf{x}}(m|m-1)$ is obtained as

$$\mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\tilde{\mathbf{x}}(m|m-1)^T] = \mathbf{A}\mathcal{E}[\tilde{\mathbf{x}}(m-1)\tilde{\mathbf{x}}(m-1)^T]\mathbf{A}^T + \mathbf{Q} \quad (7.32)$$

or

$$\mathbf{P}(m|m-1) = \mathbf{A}\mathbf{P}(m-1)\mathbf{A}^T + \mathbf{Q} \quad (7.33)$$

where $\mathbf{P}(m|m-1)$ and $\mathbf{P}(m)$ are the covariance matrices of the prediction error $\tilde{\mathbf{x}}(m|m-1)$ and estimation error $\tilde{\mathbf{x}}(m)$, respectively. A recursive relation for the covariance matrix of the signal estimation error vector, $\tilde{\mathbf{x}}(m)$, can be derived as follows. Subtracting both sides of Equation (7.18) from $\tilde{\mathbf{x}}(m)$ we have

$$\tilde{\mathbf{x}}(m) = \tilde{\mathbf{x}}(m|m-1) - \mathbf{K}(m)\mathbf{v}(m) \quad (7.34)$$

From Equation (7.34) the covariance matrix of the estimation error vector can be expressed as

$$\begin{aligned} \mathcal{E}[\tilde{\mathbf{x}}(m)\tilde{\mathbf{x}}(m)^T] &= \mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\tilde{\mathbf{x}}(m|m-1)^T] - \mathbf{K}(m)\mathcal{E}[\mathbf{v}(m)\mathbf{v}(m)^T]\mathbf{K}(m)^T \\ &\quad + 2\mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\mathbf{v}(m)^T]\mathbf{K}(m)^T \end{aligned} \quad (7.35)$$

From Equation (7.23) we have

$$\mathbf{K}(m)\mathcal{E}[\mathbf{v}(m)\mathbf{v}(m)^T] = \mathbf{P}(m|m-1)\mathbf{H}^T \quad (7.36)$$

From Equation (7.24) we have

$$\mathcal{E}[\tilde{\mathbf{x}}(m|m-1)\mathbf{v}(m)^T] = \mathbf{P}(m|m-1)\mathbf{H}^T \quad (7.37)$$

Substitution of Equations (7.36) and (7.37) in Equation (7.35) and rearranging yields

$$\mathbf{P}(m) = [\mathbf{I} - \mathbf{K}\mathbf{H}]\mathbf{P}(m|m-1) \quad (7.38)$$

Example 7.1: Recursive Estimation of a Constant Signal Observed in Noise

Consider the estimation of a constant signal observed in a random noise. The state and observation equations for this problem are given by

$$x(m) = x(m-1) = x \quad (7.39)$$

$$y(m) = x + n(m) \quad (7.40)$$

Note that $a = 1$, state excitation $e(m) = 0$, the variance of excitation $Q = 0$ and the variance of noise $R = \sigma_n^2$.

Using the Kalman algorithm, we have the following recursive solutions:

Initial conditions:

$$P(-1) = \delta \quad (7.41)$$

$$\hat{x}(0|-1) = 0 \quad (7.42)$$

for $m = 0, 1, \dots$

Time-update equations

Signal prediction equation:

$$\hat{x}(m|m-1) = \hat{x}(m-1) \quad (7.43)$$

Covariance matrix of prediction error:

$$P(m|m-1) = P(m-1) \quad (7.44)$$

Measurement-update equations

Kalman gain vector:

$$K(m) = P(m|m-1) [P(m|m-1) + \sigma_n^2]^{-1} \quad (7.45)$$

Signal estimation equation:

$$\hat{x}(m) = \hat{x}(m|m-1) + K(m) [y(m) - \hat{x}(m|m-1)] \quad (7.46)$$

Covariance matrix of estimation error:

$$P(m) = [1 - K(m)] P(m|m-1) \quad (7.47)$$

Example 7.2: Estimation of an AR Signal Observed in Noise

Consider the Kalman filtering of a P th order AR process $x(m)$ observed in an additive white Gaussian noise, $n(m)$. Assume that the signal generation and the observation equations are given as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \quad (7.48)$$

$$y(m) = x(m) + n(m) \quad (7.49)$$

Let $\sigma_e^2(m)$ and $\sigma_n^2(m)$ denote the variances of the excitation signal $e(m)$ and the noise $n(m)$, respectively. Substituting $\mathbf{A} = \mathbf{a} = [a_1, a_2, \dots, a_p]$ and $\mathbf{H} = 1$ in the Kalman filter equations yields the following Kalman filter algorithm:

Initial conditions:

$$P(-1) = \delta \quad (7.50)$$

$$\hat{x}(0|-1) = 0 \quad (7.51)$$

for $m = 0, 1, \dots$

Time-update equations

Signal prediction equation:

$$\hat{x}(m|m-1) = \sum_{k=1}^P a_k \hat{x}(m-k) \quad (7.52)$$

Covariance matrix of prediction error

$$P(m|m-1) = \mathbf{a}P(m-1)\mathbf{a}^T + \sigma_e^2 \quad (7.53)$$

Measurement-update equations

Kalman gain vector:

$$K(m) = P(m|m-1) [P(m|m-1) + \sigma_n^2]^{-1} \quad (7.54)$$

Signal estimation equation:

$$\hat{x}(m) = \hat{x}(m|m-1) + K(m) [y(m) - \hat{x}(m|m-1)] \quad (7.55)$$

Covariance matrix of estimation error:

$$P(m) = [1 - K(m)] P(m|m-1) \quad (7.56)$$

7.3 SAMPLE-ADAPTIVE FILTERS

Adaptive filters, namely the RLS, the steepest descent and the LMS, are recursive formulations of the least square error Wiener filter. Sample-adaptive filters have a number of advantages over the block-adaptive filters of Chapter 6, including lower processing delay and better tracking of the trajectory of nonstationary signals. These are essential characteristics in applications such as echo cancellation, adaptive delay estimation, low-delay predictive coding, noise cancellation, radar and channel equalization in mobile telephony, where low delay and fast tracking of time-varying processes and time-varying environments are important objectives.

Figure 7.2 illustrates the configuration of a least square error adaptive filter. At each sampling time, an adaptation algorithm adjusts the P filter coefficients $\mathbf{w}(m) = [w_0(m), w_1(m), \dots, w_{p-1}(m)]$ to minimise the difference between the filter output and

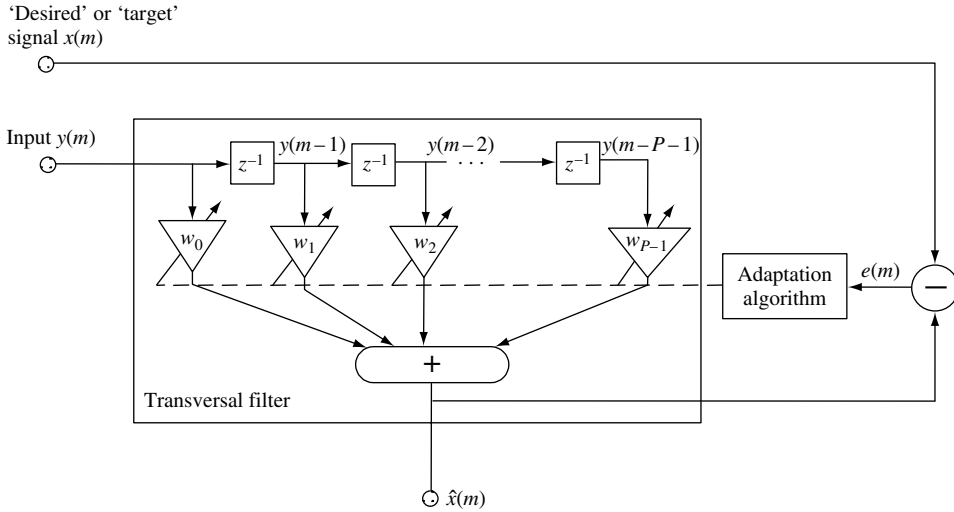


Figure 7.2 Illustration of the configuration of an adaptive filter.

a desired, or target, signal. An adaptive filter starts at some initial state, then the filter coefficients are periodically updated, usually on a sample-by-sample basis, to minimise the difference between the filter output and a desired or target signal. The adaptation formula has the general recursive form:

$$\text{Next parameter estimate} = \text{Previous parameter estimate} + \text{Update (error)}$$

where the update term is a function of the error signal. In adaptive filtering several decisions have to be made concerning the filter model and the adaptation algorithm:

- (1) *Filter type* – this can be an FIR filter, or an IIR filter. In this chapter we only consider FIR filters, since they have good stability and convergence properties and for these reasons are the type often used in practice.
- (2) *Filter order* – often the correct number of filter taps is unknown. The filter order is either set using *a priori* knowledge of the input and the desired signals, or obtained by monitoring the changes in the error signal as a function of the increasing filter order.
- (3) *Adaptation algorithm* – the two commonly used adaptation algorithms are the RLS error and the LMS methods. The factors that influence the choice of the adaptation algorithm are the computational complexity, the speed of convergence to optimal operating condition, the minimum error at convergence, the numerical stability and the robustness of the algorithm to initial parameter states.

7.4 RECURSIVE LEAST SQUARE ADAPTIVE FILTERS

The recursive least square error filter is a sample-adaptive, time-update, version of the Wiener filter studied in Chapter 6. For stationary signals, the RLS filter converges to the same optimal filter coefficients as the Wiener filter. For nonstationary signals, the RLS

filter tracks the time variations of the process. The RLS filter has a relatively fast rate of convergence to the optimal filter coefficients. This is useful in applications such as speech enhancement, channel equalization, echo cancellation and radar where the filter should be able to track relatively fast changes in the signal process.

In the recursive least square algorithm, the adaptation starts with some initial filter state, and successive samples of the input signals are used to adapt the filter coefficients. Figure 7.2 illustrates the configuration of an adaptive filter where $y(m)$, $x(m)$ and $\mathbf{w}(m) = [w_0(m), w_1(m), \dots, w_{p-1}(m)]$ denote the filter input, the desired (target) signal and the filter coefficient vector, respectively. The filter output can be expressed as

$$\hat{x}(m) = \mathbf{w}^T(m)y(m) \quad (7.57)$$

where $\hat{x}(m)$ is an estimate of the desired signal $x(m)$. The filter error signal is defined as the difference between the filter output and the target signal as

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \mathbf{w}^T(m)y(m) \end{aligned} \quad (7.58)$$

The adaptation process is based on the minimization of the mean square error criterion defined as

$$\begin{aligned} \mathcal{E}[e^2(m)] &= \mathcal{E}\left\{[x(m) - \mathbf{w}^T(m)y(m)]^2\right\} \\ &= \mathcal{E}[x^2(m)] - 2\mathbf{w}^T(m)\mathcal{E}[y(m)x(m)] + \mathbf{w}^T(m)\mathcal{E}[y(m)y^T(m)]\mathbf{w}(m) \\ &= r_{xx}(0) - 2\mathbf{w}^T(m)\mathbf{r}_{yx}(m) + \mathbf{w}^T(m)\mathbf{R}_{yy}(m)\mathbf{w}(m) \end{aligned} \quad (7.59)$$

where $r_{xx}(0)$ is the autocorrelation at lag zero of the target signal $x(m)$, \mathbf{R}_{yy} is the autocorrelation matrix of the input signal vector $\mathbf{y}(m)$ and \mathbf{r}_{yx} is the cross-correlation vector of the input and the target signals.

The Wiener filter is obtained by minimising the mean square error with respect to the filter coefficients. For stationary signals, the result of this minimisation is given in Chapter 6, Equation (6.10), as

$$\mathbf{w} = \mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} \quad (7.60)$$

In the following, we formulate a recursive, time-update, adaptive formulation of Equation (7.60). From Section 6.2, for a block of N sample vectors, the correlation matrix can be written as

$$\mathbf{R}_{yy} = \mathbf{Y}^T\mathbf{Y} = \sum_{m=0}^{N-1} \mathbf{y}(m)\mathbf{y}^T(m) \quad (7.61)$$

where $\mathbf{y}(m) = [y(m), \dots, y(m-P+1)]^T$. Now, the sum of vector product in Equation (7.61) can be expressed in recursive fashion as

$$\mathbf{R}_{yy}(m) = \mathbf{R}_{yy}(m-1) + \mathbf{y}(m)\mathbf{y}^T(m) \quad (7.62)$$

To introduce adaptability to the time variations of the signal statistics, the autocorrelation estimate in Equation (7.61) can be windowed by an exponentially decaying window:

$$\mathbf{R}_{yy}(m) = \lambda \mathbf{R}_{yy}(m-1) + \mathbf{y}(m)\mathbf{y}^T(m) \quad (7.63)$$

where λ is the so-called ‘adaptation’, or ‘forgetting’ factor, and is in the range $0 > \lambda > 1$. Similarly, the cross-correlation vector is given by

$$\mathbf{r}_{yx} = \sum_{m=0}^{N-1} \mathbf{y}(m)x(m) \quad (7.64)$$

The sum of products in Equation (7.64) can be calculated in recursive form as

$$\mathbf{r}_{yx}(m) = \mathbf{r}_{yx}(m-1) + \mathbf{y}(m)x(m) \quad (7.65)$$

Equation (7.65) can be made adaptive using an exponentially decaying forgetting factor λ :

$$\mathbf{r}_{yx}(m) = \lambda \mathbf{r}_{yx}(m-1) + \mathbf{y}(m)x(m) \quad (7.66)$$

For a recursive solution of the least square error Equation (7.66), we need to obtain a recursive time-update formula for the inverse matrix in the form

$$\mathbf{R}_{yy}^{-1}(m) = \mathbf{R}_{yy}^{-1}(m-1) + \text{Update}(m) \quad (7.67)$$

A recursive relation for the matrix inversion is obtained using the following lemma.

7.4.1 THE MATRIX INVERSION LEMMA

Let \mathbf{A} and \mathbf{B} be two positive-definite $P \times P$ matrices related by

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T \quad (7.68)$$

where \mathbf{D} is a positive-definite $N \times N$ matrix and \mathbf{C} is a $P \times N$ matrix. The matrix inversion lemma states that the inverse of the matrix \mathbf{A} can be expressed as

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B} \quad (7.69)$$

This lemma can be proved by multiplying Equation (7.68) and Equation (7.69). The left- and right-hand sides of the results of multiplication are the identity matrix. The matrix inversion lemma can be used to obtain a recursive implementation for the inverse of the correlation matrix, $\mathbf{R}_{yy}^{-1}(m)$. Let

$$\mathbf{R}_{yy}(m) = \mathbf{A} \quad (7.70)$$

$$\lambda^{-1}\mathbf{R}_{yy}^{-1}(m-1) = \mathbf{B} \quad (7.71)$$

$$\mathbf{y}(m) = \mathbf{C} \quad (7.72)$$

$$\mathbf{D} = \text{identity matrix} \quad (7.73)$$

Substituting Equations (7.63) and (7.64) in Equation (7.62), we obtain

$$\mathbf{R}_{yy}^{-1}(m) = \lambda^{-1}\mathbf{R}_{yy}^{-1}(m-1) - \frac{\lambda^{-2}\mathbf{R}_{yy}^{-1}(m-1)\mathbf{y}(m)\mathbf{y}^T(m)\mathbf{R}_{yy}^{-1}(m-1)}{1 + \lambda^{-1}\mathbf{y}^T(m)\mathbf{R}_{yy}^{-1}(m-1)\mathbf{y}(m)} \quad (7.74)$$

Now define the variables $\Phi(m)$ and $\mathbf{k}(m)$ as

$$\Phi_{yy}(m) = \mathbf{R}_{yy}^{-1}(m) \quad (7.75)$$

and

$$\mathbf{k}(m) = \frac{\lambda^{-1}\mathbf{R}_{yy}^{-1}(m-1)\mathbf{y}(m)}{1 + \lambda^{-1}\mathbf{y}^T(m)\mathbf{R}_{yy}^{-1}(m-1)\mathbf{y}(m)} \quad (7.76)$$

or

$$\mathbf{k}(m) = \frac{\lambda^{-1}\Phi_{yy}(m-1)\mathbf{y}(m)}{1 + \lambda^{-1}\mathbf{y}^T(m)\Phi_{yy}(m-1)\mathbf{y}(m)} \quad (7.77)$$

Using Equations (7.75) and (7.76), the recursive equation (7.67) for computing the inverse matrix can be written as

$$\Phi_{yy}(m) = \lambda^{-1}\Phi_{yy}(m-1) - \lambda^{-1}\mathbf{k}(m)\mathbf{y}^T(m)\Phi_{yy}(m-1) \quad (7.78)$$

From Equations (7.77) and (7.78), we have

$$\begin{aligned} \mathbf{k}(m) &= [\lambda^{-1}\Phi_{yy}(m-1) - \lambda^{-1}\mathbf{k}(m)\mathbf{y}^T(m)\Phi_{yy}(m-1)]\mathbf{y}(m) \\ &= \Phi_{yy}(m)\mathbf{y}(m) \end{aligned} \quad (7.79)$$

Now Equations (7.78) and (7.79) are used in the following to derive the RLS adaptation algorithm.

7.4.2 RECURSIVE TIME-UPDATE OF FILTER COEFFICIENTS

The least square error filter coefficients are

$$\begin{aligned} \mathbf{w}(m) &= \mathbf{R}_{yy}^{-1}(m)\mathbf{r}_{yx}(m) \\ &= \Phi_{yy}(m)\mathbf{r}_{yx}(m) \end{aligned} \quad (7.80)$$

Substituting the recursive form of the correlation vector in Equation (7.80) yields

$$\begin{aligned} \mathbf{w}(m) &= \Phi_{yy}(m) [\lambda\mathbf{r}_{yx}(m-1) + \mathbf{y}(m)x(m)] \\ &= \lambda\Phi_{yy}(m)\mathbf{r}_{yx}(m-1) + \Phi_{yy}(m)\mathbf{y}(m)x(m) \end{aligned} \quad (7.81)$$

Now substitution of the recursive form of the matrix $\Phi_{yy}(m)$ from Equation (7.78) and $\mathbf{k}(m) = \Phi(m)\mathbf{y}(m)$ from Equation (7.79) in the right-hand side of Equation (7.81) yields

$$\mathbf{w}(m) = [\lambda^{-1}\Phi_{yy}(m-1) - \lambda^{-1}\mathbf{k}(m)\mathbf{y}^T(m)\Phi_{yy}(m-1)]\lambda\mathbf{r}_{yx}(m-1) + \mathbf{k}(m)x(m) \quad (7.82)$$

or

$$\mathbf{w}(m) = \Phi_{yy}(m-1)\mathbf{r}_{yx}(m-1) - \mathbf{k}(m)\mathbf{y}^T(m)\Phi_{yy}(m-1)\mathbf{r}_{yx}(m-1) + \mathbf{k}(m)x(m) \quad (7.83)$$

Substitution of $\mathbf{w}(m-1) = \Phi(m-1)\mathbf{r}_{yx}(m-1)$ in Equation (7.83) yields

$$\mathbf{w}(m) = \mathbf{w}(m-1) + \mathbf{k}(m)[x(m) - \mathbf{y}^T(m)\mathbf{w}(m-1)] \quad (7.84)$$

This equation can be rewritten in the following form

$$\mathbf{w}(m) = \mathbf{w}(m-1) - \mathbf{k}(m)e(m) \quad (7.85)$$

Equation (7.85) is a recursive time-update implementation of the least square error Wiener filter.

RLS Adaptation Algorithm

Input signals: $y(m)$ and $x(m)$

Initial values:

$$\Phi_{yy}(m) = \delta\mathbf{I}$$

$$\mathbf{w}(0) = \mathbf{w}_1$$

For $m = 1, 2, \dots$

Filter gain vector update:

$$\mathbf{k}(m) = \frac{\lambda^{-1}\Phi_{yy}(m-1)\mathbf{y}(m)}{1 + \lambda^{-1}\mathbf{y}^T(m)\Phi_{yy}(m-1)\mathbf{y}(m)} \quad (7.86)$$

Error signal equation:

$$e(m) = x(m) - \mathbf{w}^T(m-1)\mathbf{y}(m) \quad (7.87)$$

Filter coefficients adaptation:

$$\mathbf{w}(m) = \mathbf{w}(m-1) + \mathbf{k}(m)e(m) \quad (7.88)$$

Inverse correlation matrix update:

$$\Phi_{yy}(m) = \lambda^{-1}\Phi_{yy}(m-1) - \lambda^{-1}\mathbf{k}(m)\mathbf{y}^T(m)\Phi_{yy}(m-1) \quad (7.89)$$

7.5 THE STEEPEST-DESCENT METHOD

The surface of the mean square output error of an adaptive FIR filter, with respect to the filter coefficients, is a quadratic bowl-shaped curve, with a single global minimum that corresponds to the LSE filter coefficients.

Figure 7.3 illustrates the mean square error curve for a single coefficient filter. This figure also illustrates the steepest-descent search for the minimum mean square error coefficient. The search is based on taking a number of successive downward steps in the direction of negative gradient of the error surface. Starting with a set of initial values, the filter coefficients are successively updated in the downward direction, until the minimum point, at which the gradient is zero, is reached. The steepest-descent adaptation method can be expressed as

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \mu \left[-\frac{\partial \mathcal{E}[e^2(m)]}{\partial \mathbf{w}(m)} \right] \quad (7.90)$$

where μ is the adaptation step size.

From Equation (7.59), the gradient of the mean square error function is given by

$$\frac{\partial \mathcal{E}[e^2(m)]}{\partial \mathbf{w}(m)} = -2\mathbf{r}_{yx} + 2\mathbf{R}_{yy}\mathbf{w}(m) \quad (7.91)$$

Substituting Equation (7.91) in Equation (7.90) yields

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \mu [\mathbf{r}_{yx} - \mathbf{R}_{yy}\mathbf{w}(m)] \quad (7.92)$$

where the factor of 2 in Equation (7.91) has been absorbed in the adaptation step size μ . Let \mathbf{w}_o denote the optimal LSE filter coefficient vector; we define a filter coefficients error vector $\tilde{\mathbf{w}}(m)$ as

$$\tilde{\mathbf{w}}(m) = \mathbf{w}(m) - \mathbf{w}_o \quad (7.93)$$

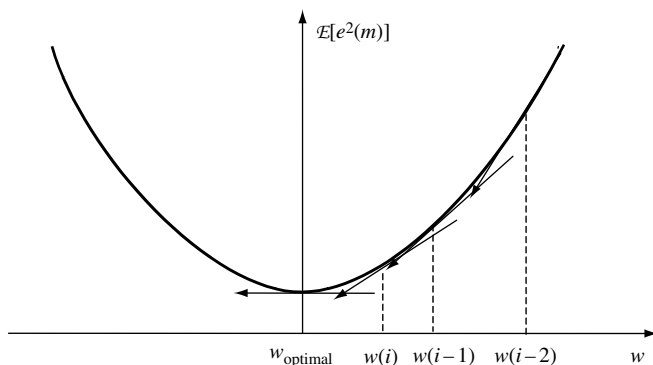


Figure 7.3 Illustration of gradient search of the mean square error surface for the minimum error point.

For a stationary process, the optimal LSE filter \mathbf{w}_o is obtained from the Wiener filter, Equation (5.10), as

$$\mathbf{w}_o = \mathbf{R}_{yy}^{-1} \mathbf{r}_{yx} \quad (7.94)$$

Subtracting \mathbf{w}_o from both sides of Equation (7.92), and then substituting $\mathbf{R}_{yy} \mathbf{w}_o$ for \mathbf{r}_{yx} and using Equation (7.93) yields

$$\tilde{\mathbf{w}}(m+1) = [\mathbf{I} - \mu \mathbf{R}_{yy}] \tilde{\mathbf{w}}(m) \quad (7.95)$$

It is desirable that the filter error vector, $\tilde{\mathbf{w}}(m)$, vanishes as rapidly as possible. The parameter μ , the adaptation step size, controls the stability and the rate of convergence of the adaptive filter. Too large a value for μ causes instability; too small a value gives a low convergence rate. The stability of the parameter estimation method depends on the choice of the adaptation parameter, μ , and the autocorrelation matrix.

From Equation (7.95), a recursive equation for the error in each individual filter coefficient can be obtained as follows. The correlation matrix can be expressed in terms of the matrices of eigenvectors and eigenvalues as

$$\mathbf{R}_{yy} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \quad (7.96)$$

where \mathbf{Q} is an orthonormal matrix of the eigenvectors of \mathbf{R}_{yy} , and $\mathbf{\Lambda}$ is a diagonal matrix with its diagonal elements corresponding to the eigenvalues of \mathbf{R}_{yy} . Substituting \mathbf{R}_{yy} from Equation (7.96) in Equation (7.95) yields

$$\tilde{\mathbf{w}}(m+1) = [\mathbf{I} - \mu \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T] \tilde{\mathbf{w}}(m) \quad (7.97)$$

Multiplying both sides of Equation (7.97) by \mathbf{Q}^T and using the relation $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q} \mathbf{Q}^T = \mathbf{I}$ yields

$$\mathbf{Q}^T \tilde{\mathbf{w}}(m+1) = [\mathbf{I} - \mu \mathbf{\Lambda}] \mathbf{Q}^T \tilde{\mathbf{w}}(m) \quad (7.98)$$

Let

$$\mathbf{v}(m) = \mathbf{Q}^T \tilde{\mathbf{w}}(m) \quad (7.99)$$

Then

$$\mathbf{v}(m+1) = [\mathbf{I} - \mu \mathbf{\Lambda}] \mathbf{v}(m) \quad (7.100)$$

As $\mathbf{\Lambda}$ and \mathbf{I} are both diagonal matrices, Equation (7.92) can be expressed in terms of the equations for the individual elements of the error vector, $\mathbf{v}(m)$, as

$$v_k(m+1) = [1 - \mu \lambda_k] v_k(m) \quad (7.101)$$

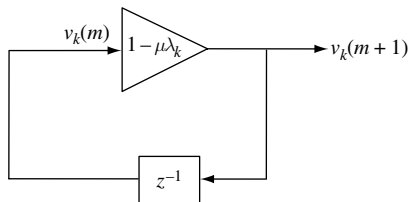


Figure 7.4 A feedback model of the variation of coefficient error with time.

where λ_k is the k th eigenvalue of the autocorrelation matrix of the filter input, $y(m)$. Figure 7.4 is a feedback network model of the time variations of the error vector. From Equation (7.101), the condition for the stability of the adaptation process and the decay of the coefficient error vector is

$$-1 < 1 - \mu\lambda_k < 1 \quad (7.102)$$

Let λ_{\max} denote the maximum eigenvalue of the autocorrelation matrix of $y(m)$ then, from Equation (7.102), the limits on μ for stable adaptation are given by

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (7.103)$$

7.5.1 CONVERGENCE RATE

The convergence rate of the filter coefficients depends on the choice of the adaptation step size μ , where $0 < \mu < 2/\lambda_{\max}$. When the eigenvalues of the correlation matrix are unevenly spread, the filter coefficients converge at different speeds: the smaller the k th eigenvalue, the slower the speed of convergence of the k th coefficients. The filter coefficients with maximum and minimum eigenvalues, λ_{\max} and λ_{\min} , converge according to the following equations:

$$v_{\max}(m+1) = (1 - \mu\lambda_{\max}) v_{\max}(m) \quad (7.104)$$

$$v_{\min}(m+1) = (1 - \mu\lambda_{\min}) v_{\min}(m) \quad (7.105)$$

The ratio of the maximum to the minimum eigenvalue of a correlation matrix is called the eigenvalue spread of the correlation matrix:

$$\text{eigenvalue spread} = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (7.106)$$

Note that the spread in the speed of convergence of filter coefficients is proportional to the spread in eigenvalue of the autocorrelation matrix of the input signal.

Example 7.3

Assuming that the maximum eigenvalue of a signal is 2 and the minimum eigenvalue is 0.2, calculate:

- (1) the eigenvalue spread;
- (2) the bounds on adaptation step size;
- (3) the decay factor of the error equations for the fastest and the slowest converging coefficients of the filter, assuming that the adaptation step size is 0.4.

Answer:

- (1) eigenvalue spread =

$$\frac{\lambda_{\max}}{\lambda_{\min}} = 10$$

- (2) the bounds on adaptation step size,

$$0 < \mu < \frac{2}{\lambda_{\max}} = 1$$

- (3) the fastest decay factor = $(1 - \mu\lambda_{\max}) = 1 - 0.4 \times 2 = 0.2$, and the slowest decay factor = $(1 - \mu\lambda_{\min}) = 1 - 0.4 \times 0.2 = 0.92$.

7.5.2 VECTOR-VALUED ADAPTATION STEP SIZE

Instead of using a single scalar-valued adaptation step size, μ , we can use a vector-valued adaptation step size, $\boldsymbol{\mu} = [\mu_0, \mu_2, \dots, \mu_{p-1}]$, with each filter coefficient, w_k , having its own adaptation step size, μ_k . This is useful when the input signal has an eigenvalue spread of greater than one, in which case, as shown in the preceding example, the use of a single step adaptation size would cause an uneven rate of convergence of coefficients. With the use of a vector-valued step size, the k th adaptation step size can be adjusted using the k th eigenvalue of the autocorrelation matrix of the input signal to the filter to ensure a more even rate of convergence of different filter coefficients.

7.6 THE LMS FILTER

In its search for the least square error filter coefficients, the steepest-descent method employs the gradient of the *averaged* squared error. A computationally simpler version of the gradient search method is the LMS filter, in which the gradient of the *mean* square error is substituted with the gradient of the *instantaneous* squared error function. The LMS adaptation method is defined as

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \mu \left[-\frac{\partial e^2(m)}{\partial \mathbf{w}(m)} \right] \quad (7.107)$$

where the error signal $e(m)$ is the difference between the adaptive filter output and the target (desired) signal $x(m)$, given by

$$e(m) = x(m) - \mathbf{w}^T(m)\mathbf{y}(m) \quad (7.108)$$

The instantaneous gradient of the squared error can be re-expressed as

$$\begin{aligned} \frac{\partial e^2(m)}{\partial \mathbf{w}(m)} &= \frac{\partial}{\partial \mathbf{w}(m)} [x(m) - \mathbf{w}^T(m)\mathbf{y}(m)]^2 \\ &= -2\mathbf{y}(m)[x(m) - \mathbf{w}^T(m)\mathbf{y}(m)] \\ &= -2\mathbf{y}(m)e(m) \end{aligned} \quad (7.109)$$

Substituting Equation (7.109) into the recursion filter update Equation (7.107) yields the LMS adaptation equation:

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \mu [\mathbf{y}(m)e(m)] \quad (7.110)$$

It can be seen that the filter update equation is very simple. The LMS filter is widely used in adaptive filter applications such as adaptive equalisation, echo cancellation, radar, etc. The main advantage of the LMS algorithm is its simplicity both in terms of the memory requirement and the computational complexity, which is $O(P)$, where P is the filter length.

7.6.1 LEAKY LMS ALGORITHM

The stability and the adaptability of the recursive LMS adaptation Equation (7.86) can be improved by introducing a so-called leakage factor α as

$$\mathbf{w}(m+1) = \alpha \mathbf{w}(m) + \mu [\mathbf{y}(m)e(m)] \quad (7.111)$$

Note that, as illustrated in Figure 7.5, the feedback equation for the time update of the filter coefficients is essentially a recursive (infinite impulse response) system with input $\mu \mathbf{y}(m)e(m)$ and its poles at α . When the parameter $\alpha < 1$, the effect is to introduce more stability and accelerate the filter adaptation to the changes in input signal characteristics.

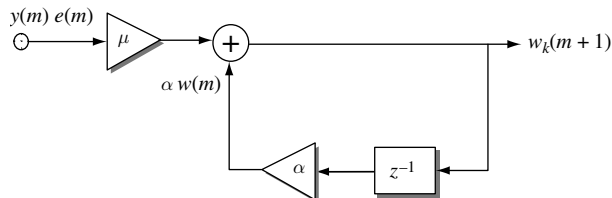


Figure 7.5 Illustration of leaky LMS adaptation of a filter coefficient.

7.6.2 NORMALISED LMS ALGORITHM

The normalised LMS adaptation equation is given by

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \frac{\mu}{a + \sum_{k=0}^M y^2(m-k)} \mathbf{y}(m)e(m) \quad (7.112)$$

where $\sum_{k=0}^M y^2(m-k)$ is the input signal energy, μ controls adaptation step size and a is a small constant employed to avoid the denominator of the update term becoming zero when the input signal $\mathbf{y}(m)$ is zero.

7.6.2.1 Derivation of the Normalised LMS Algorithm

In normalised LMS, instead of using the LMS criterion of minimising the difference between the filter output and the desired output, the criterion of minimising the Euclidean norm of incremental change, $\delta\mathbf{w}(m+1)$, in successive updates of the filter coefficient vector is used:

$$\|\delta\mathbf{w}(m+1)\| = \sum_{k=0}^{P-1} [w_k(m+1) - w_k(m)]^2 \quad (7.113)$$

Subject to the constraint that

$$\mathbf{w}^T(m+1)\mathbf{y}(m) = x(m) \quad (7.114)$$

The solution that satisfies the above criterion and the constraint can be obtained by the Lagrange multipliers method.

In the Lagrange method we define an optimization criterion $J(\cdot)$ as a combination of the criterion expressed in Equation (7.113) and the constraint expressed in Equation (7.114) as

$$\begin{aligned} J[\mathbf{w}(m+1), \lambda] &= \|\delta\mathbf{w}(m+1)\| + \lambda[x(m) - \mathbf{w}^T(m+1)\mathbf{y}(m)] \\ &= \sum_{k=0}^{P-1} [w_k(m+1) - w_k(m)]^2 - \lambda[x(m) \\ &\quad - \sum_{k=0}^{P-1} w_k(m+1)y(m-k)] \end{aligned} \quad (7.115)$$

To obtain the minimum of the criterion $J[\mathbf{w}(m+1), \lambda]$, set the derivative of J with respect to each coefficient $w_i(m+1)$ to zero as

$$\frac{\partial J[\mathbf{w}(m+1), \lambda]}{\partial w_i(m+1)} = 2w_i(m+1) - 2w_i(m) - \lambda y(m-i) = 0 \quad i = 0, \dots, P-1 \quad (7.116)$$

From Equation (7.116) we have

$$w_i(m+1) = w_i(m) - \frac{\lambda}{2} y(m-i) \quad (7.117)$$

From Equations (7.117) and (7.114) we have

$$\sum_{m=0}^{P-1} w_i(m)y(m-i) - \frac{\lambda}{2} \sum_{m=0}^{P-1} y(m-i)^2 = x(m) \quad (7.118)$$

Hence the Lagrange parameter λ is given by

$$\lambda = \frac{2[x(m) - \sum_{i=0}^{P-1} w_i(m)y(m-i)]}{\sum_{i=0}^{P-1} y^2(m-i)} = \frac{2e(m)}{\sum_{i=0}^{P-1} y^2(m-i)} \quad (7.119)$$

Substitution for λ from Equation (7.119) in Equation (7.117) yields the normalised LMS (NLMS) equation

$$w_i(m+1) = w_i(m) - \frac{e(m)}{\sum_{k=0}^{P-1} y^2(m-k)} y(m-i) \quad (7.120)$$

The NLMS Equation (7.112) is obtained from Equation (7.120) by introducing a variable μ to control the step size and a variable a to prevent the denominator of the update term, Equation (7.120), becoming zero when the input signal vector, y , has zero values.

7.6.2.2 Steady-state Error in LMS

The optimal LSE, E_{\min} , is achieved when the filter coefficients approach the optimum value defined by the block least square error equation $\mathbf{w}_o = \mathbf{R}_{yy}^{-1} \mathbf{r}_{yx}$, derived in Chapter 6. The steepest-descent method employs the average gradient of the error surface for incremental updates of the filter coefficients towards the optimal value. Hence, when the filter coefficients reach the minimum point of the mean square error curve, the *averaged* gradient is zero and will remain zero so long as the error surface is stationary. In contrast, examination of the LMS equation shows that, for applications in which the LSE is nonzero, such as noise reduction, the incremental update term, $\mu e(m)y(m)$, would remain nonzero even when the optimal point was reached. Thus, at the convergence, the LMS filter will randomly vary about the LSE point, with the result that the LSE for the LMS will be in excess of the LSE for Wiener or steepest-descent methods. Note that at, or near, convergence, a gradual decrease in μ would decrease the excess LSE at the expense of some loss of adaptability to changes in the signal characteristics.

7.7 SUMMARY

This chapter began with an introduction to Kalman filter theory. The Kalman filter was derived using the orthogonality principle: for the optimal filter, the innovation sequence must be an uncorrelated process and orthogonal to the past observations. Note that the same principle can also be used to derive the Wiener filter coefficients. Although, like the

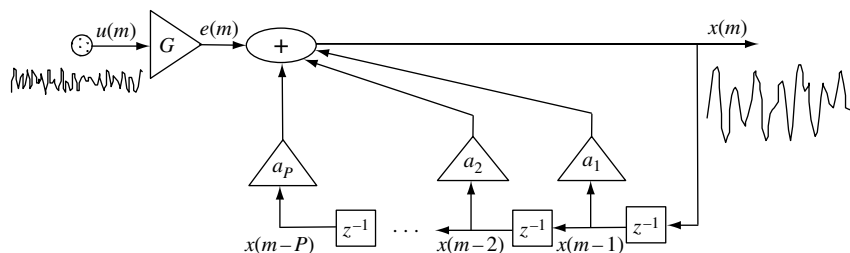
Wiener filter, the derivation of the Kalman filter is based on the LSE criterion, the Kalman filter differs from the Wiener filter in two respects. First, the Kalman filter can be applied to nonstationary processes and, second, the Kalman theory employs a model of the signal generation process in the form of the state equation. This is an important advantage in the sense that the Kalman filter can be used to explicitly model the dynamics of the signal process.

For many practical applications, such as echo cancellation, channel equalisation, adaptive noise cancellation and time-delay estimation, the RLS and LMS filters provide a suitable alternative to the Kalman filter. The RLS filter is a recursive implementation of the Wiener filter and, for stationary processes, it should converge to the same solution as the Wiener filter. The main advantage of the LMS filter is the relative simplicity of the algorithm. However, for signals with a large spectral dynamic range, or equivalently a large eigenvalue spread, the LMS has an uneven and slow rate of convergence. If, in addition to having a large eigenvalue spread, a signal is also nonstationary (e.g. speech and audio signals), then the LMS can be an unsuitable adaptation method, and the RLS method, with its better convergence rate and lower sensitivity to the eigenvalue spread, becomes a more attractive alternative.

BIBLIOGRAPHY

- Alexander S.T. (1986) *Adaptive Signal Processing: Theory and Applications*. Springer, New York.
- Bellanger M.G. (1988) *Adaptive Filters and Signal Analysis*. Marcel Dekker, New York.
- Bershad N.J. (1986) Analysis of the normalised LMS algorithm with Gaussian inputs. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-34**: 793–807.
- Bershad N.J. and Qu L.Z. (1989) On the probability density function of the LMS adaptive filter weights. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-37**: 43–57.
- Cioffi J.M. and Kailath T. (1984) Fast recursive least squares transversal filters for adaptive filtering. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-32**: 304–337.
- Classen T.A. and Mecklanbrauker W.F. (1985) Adaptive techniques for signal processing in communications. *IEEE Commun.*, **23**: 8–17.
- Cowan C.F. and Grant P.M. (1985) *Adaptive Filters*. Prentice-Hall, Englewood Cliffs, NJ.
- Eweda E. and Macchi O. (1985) Tracking error bounds of adaptive non-stationary filtering. *Automatica*, **21**: 293–302.
- Gabor D., Wilby W. P. and Woodcock R. (1960) A universal nonlinear filter, predictor and simulator which optimises itself by a learning process. *IEE Proc.*, **108**: 422–438.
- Gabriel W.F. (1976) Adaptive arrays: an introduction. *Proc. IEEE*, **64**: 239–272.
- Haykin S. (2002) *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ.
- Honig M.L. and Messerschmitt D.G. (1984) *Adaptive Filters: Structures, Algorithms and Applications*. Kluwer, Boston, MA.
- Kailath T. (1970) The innovations approach to detection and estimation theory. *Proc. IEEE*, **58**: 680–965.
- Kalman R.E. (1960) A new approach to linear filtering and prediction problems. *Trans. ASME, Ser. D, J. Basic Engng.* **82**: 34–45.
- Kalman R.E. and Bucy R.S. (1961) New results in linear filtering and prediction theory. *Trans. ASME, Ser. D, J. Basic Engng.* **83**: 95–108.
- Widrow B. (1990) Thirty years of adaptive neural networks: perceptron, madaline, and back propagation. *Proc. IEEE* (Special Issue on Neural Networks), **I**: 78.
- Widrow B. and Sterns S.D. (1985) *Adaptive Signal Processing*. Prentice Hall, Englewood Cliffs, NJ.
- Wilkinson J.H. (1965) *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford.
- Zadeh L.A. and Desoer C.A. (1963) *Linear System Theory: the State-space Approach*. McGraw-Hill, New York.

8



Linear Prediction Models

8.1 Linear Prediction Coding	210
8.2 Forward, Backward and Lattice Predictors	219
8.3 Short- and Long-term Predictors	226
8.4 MAP Estimation of Predictor Coefficients	228
8.5 Formant-tracking LP Models	230
8.6 Sub-band Linear Prediction Model	232
8.7 Signal Restoration Using Linear Prediction Models	233
8.8 Summary	238
Bibliography	238

Linear prediction (LP) model predicts the future value of a signal from a linear combination of its past values. LP models are used in a diverse applications, such as data forecasting, speech coding, video coding, speech recognition, model-based spectral analysis, model-based signal interpolation, signal restoration, noise reduction, impulse detection and change detection. In the statistical literature, linear prediction models are often referred to as autoregressive (AR) processes. In this chapter, we introduce the theory of linear prediction models and consider efficient methods for the computation of predictor coefficients. We study the forward, backward and lattice predictors, and consider various methods for the formulation and calculation of predictor coefficients, including the least square error and maximum *a posteriori* methods. For modelling quasi-periodic signals, such as voiced speech, an extended linear predictor that simultaneously utilizes the short- and long-term correlation structures is introduced. We study sub-band linear predictors that are particularly useful for sub-band coding and processing of noisy signals. Finally, the application of linear

prediction models in signal coding, enhancement, pattern recognition and watermarking is considered.

8.1 LINEAR PREDICTION CODING

In terms of usefulness and application as a signal processing tool, linear prediction models rank almost alongside the Fourier transform. Indeed, modern digital mobile phones employ voice coders based on linear prediction modelling of speech for efficient coding.

There are two main motivations for the use of predictors in signal processing applications:

- (1) to predict the trajectory of a signal;
- (2) to remove the predictable part of a signal in order to avoid retransmitting parts of a signal that can be predicted at the receiver and thereby save storage, bandwidth, time and power.

The accuracy with which a signal can be predicted from its past samples depends on the autocorrelation function, or equivalently the bandwidth and the power spectrum, of the signal. As illustrated in Figure 8.1, in the time domain a predictable signal has a smooth and correlated fluctuation, and in the frequency domain the energy of a predictable signal is concentrated in narrow band(s) of frequencies. In contrast, the energy of an unpredictable signal, such as a white noise, is spread over a wide band of frequencies.

For a signal to have a capacity to convey information it must have a degree of randomness. Most signals, such as speech, music and video signals, are partially predictable and partially random. These signals can be modelled as the output of a linear filter excited by an uncorrelated random input. The random input models the unpredictable part of the signal, whereas the filter models the predictable structure of the signal. The aim of linear prediction is to model the mechanism that introduces the correlation in a signal.

Linear prediction models are extensively used in speech processing applications such as in low bit-rate speech coders, speech enhancement and speech recognition. Speech is generated by inhaling air and then exhaling it through the glottis and the vocal tract. The noise-like air, from the lung, is modulated and shaped by vibrations of the glottal cords and the resonance

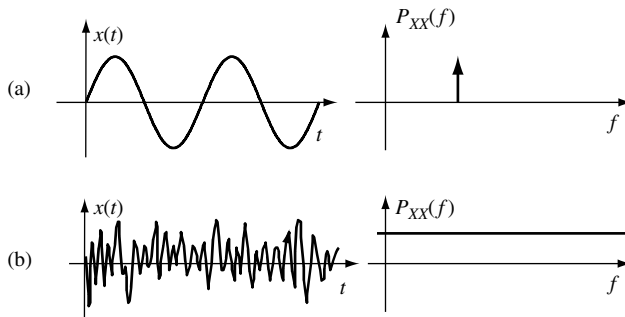


Figure 8.1 The concentration or spread of power in frequency indicates the predictable or random character of a signal: (a) a predictable signal; (b) a random signal.

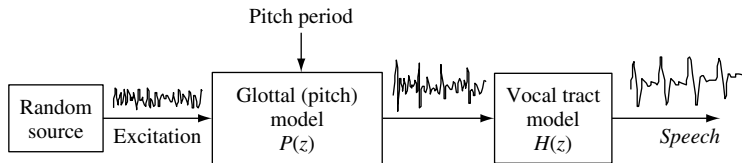


Figure 8.2 A source-Filter model of speech production. The filter is usually modelled with a linear prediction model.

of the vocal tract. Figure 8.2 illustrates a source-filter model of speech. The source models the lung, and emits a random input excitation signal that is filtered by a pitch filter. The pitch filter models the vibrations of the glottal cords, and generates a sequence of quasiperiodic excitation pulses for voiced sounds as shown in Figure 8.2. The pitch filter model is also termed the ‘long-term predictor’ since it models the correlation of each sample with the samples a pitch period away. The main source of correlation and power amplification in speech is the vocal tract. The vocal tract is modelled by a linear predictor model, which is also termed the ‘short-term predictor’, as it models the correlation of each sample with the few (typically eight to 20) preceding samples. In this section, we study the short-term linear prediction model. In Section 8.3, the predictor model is extended to include long-term pitch period correlations.

A linear predictor model, Figures 8.3 and 8.4, forecasts the amplitude of a signal at time m , $x(m)$, using a linearly weighted combination of P past samples [$x(m-1)$, $x(m-2)$, \dots , $x(m-P)$] as

$$\hat{x}(m) = \sum_{k=1}^P a_k x(m-k) \tag{8.1}$$

where the integer variable m is the discrete-time index, $\hat{x}(m)$ is the prediction of $x(m)$, and a_k are the predictor coefficients. A block diagram implementation of the predictor equation [Equation (8.1)] is illustrated in Figure 8.4.

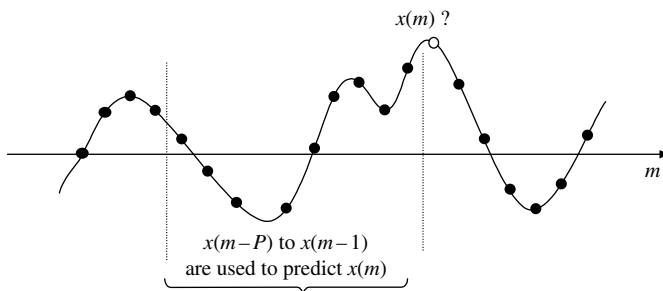


Figure 8.3 Illustration of prediction of a sample from past samples.

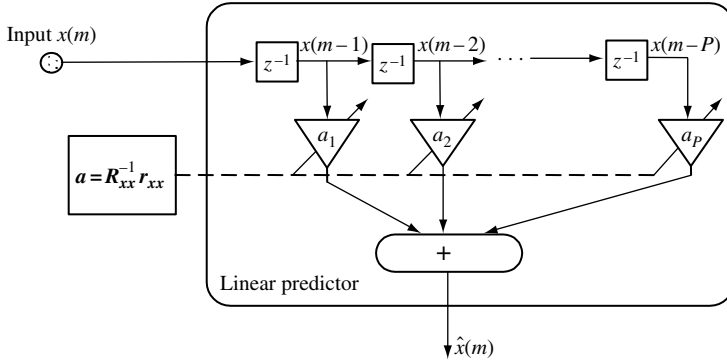


Figure 8.4 Block-diagram illustration of a linear predictor.

The prediction error $e(m)$, defined as the difference between the actual sample value $x(m)$ and its predicted value $\hat{x}(m)$, is given by

$$\begin{aligned}
 e(m) &= x(m) - \hat{x}(m) \\
 &= x(m) - \sum_{k=1}^P a_k x(m-k)
 \end{aligned}
 \tag{8.2}$$

For information-bearing signals, the prediction error $e(m)$ may be regarded as the information, or the innovation (i.e. ‘new’), content of the sample $x(m)$. From Equation (8.2) a signal generated, or modelled, by a linear predictor can be described by the following feedback equation:

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m)
 \tag{8.3}$$

Figure 8.5 illustrates a linear predictor model of a signal $x(m)$. In this model, the random input excitation (i.e. the prediction error) is $e(m) = Gu(m)$, where $u(m)$ is a zero-mean,

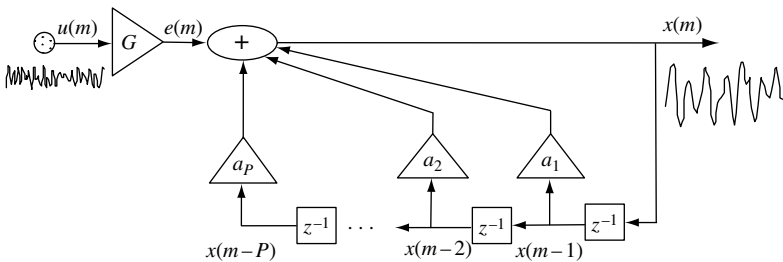


Figure 8.5 Illustration of a signal generated by a linear predictive model.

unit-variance random signal, and G , a gain term, is the square root of the variance (i.e. power) of $e(m)$:

$$G = \{\mathcal{E}[e^2(m)]\}^{1/2} \tag{8.4}$$

where $\mathcal{E}[\cdot]$ is an averaging, or expectation, operator.

8.1.1 FREQUENCY RESPONSE OF LP MODELS

The z -transform of the LP Equation (8.3) shows that the LP model is an all-pole digital filter with transfer function

$$H(z) = \frac{X(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^P a_k z^{-k}} = G \frac{1}{\prod_{k=1}^N (1 - r_k z^{-1})} \frac{1}{\prod_{k=1}^M (1 - 2r_k \cos \varphi_k z^{-1} + r_k^2 z^{-2})} \tag{8.5}$$

In Equation (8.5) it is assumed there are M complex pole pairs and N real poles with $P = N + 2M$ and r_k and φ_k the radius and angle of the k th pole. The frequency response of an LP model is given by

$$H(f) = \frac{G}{1 - \sum_{k=1}^P a_k e^{-j2\pi k f}} = G \frac{1}{\prod_{k=1}^N (1 - r_k e^{-j2\pi f})} \frac{1}{\prod_{k=1}^M (1 - 2r_k \cos \varphi_k e^{-j2\pi f} + r_k^2 e^{-j4\pi f})} \tag{8.6}$$

Figure 8.6 illustrates the relation between the poles and the magnitude frequency response of an all-pole filter. The main features of the spectral resonance at a pole are the frequency, bandwidth and magnitude of the resonance. The roots of a complex pair of poles can be written in terms of the radius r_k and the angle φ_k of the pole as

$$z_k = r_k e^{\pm j\varphi_k} \tag{8.7}$$

The resonance frequency of a complex pair of poles is

$$F(\varphi_k) = \frac{F_s}{2\pi} \varphi_k \tag{8.8}$$

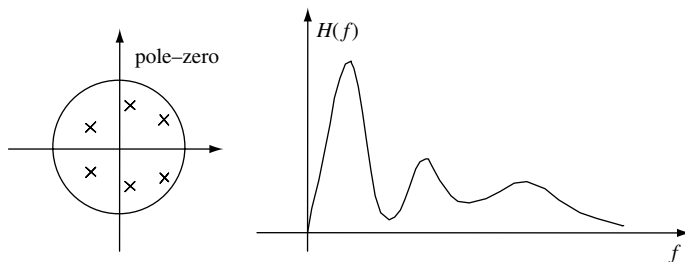


Figure 8.6 The pole-zero position and frequency response of a linear predictor.

Where F_s is the sampling frequency. The bandwidth of a pole is related to its radius r_k as

$$B_k = (-\log r_k)(F_s/\pi) \quad (8.9)$$

The magnitude of the spectral resonance at the pole is given by $H(\varphi)$.

8.1.2 CALCULATION OF PREDICTOR COEFFICIENTS

Linear predictor coefficients are obtained by minimising the mean square prediction error as

$$\begin{aligned} \mathcal{E}[e^2(m)] &= \mathcal{E} \left\{ \left[x(m) - \sum_{k=1}^P a_k x(m-k) \right]^2 \right\} \\ &= \mathcal{E}[x^2(m)] - 2 \sum_{k=1}^P a_k \mathcal{E}[x(m)x(m-k)] + \sum_{k=1}^P a_k \sum_{j=1}^P a_j \mathcal{E}[x(m-k)x(m-j)] \\ &= r_{xx}(0) - 2\mathbf{r}_{xx}^T \mathbf{a} + \mathbf{a}^T \mathbf{R}_{xx} \mathbf{a} \end{aligned} \quad (8.10)$$

where $\mathbf{R}_{xx} = \mathcal{E}[\mathbf{x}\mathbf{x}^T]$ is the autocorrelation matrix of the input vector $\mathbf{x}^T = [x(m-1), x(m-2), \dots, x(m-P)]$, $\mathbf{r}_{xx} = \mathcal{E}[x(m)\mathbf{x}]$ is the autocorrelation vector and $\mathbf{a}^T = [a_1, a_2, \dots, a_P]$ is the predictor coefficient vector. From Equation (8.10), the gradient of the mean square prediction error with respect to the predictor coefficient vector \mathbf{a} is given by

$$\frac{\partial}{\partial \mathbf{a}} \mathcal{E}[e^2(m)] = -2\mathbf{r}_{xx}^T + 2\mathbf{a}^T \mathbf{R}_{xx} \quad (8.11)$$

where the gradient vector is defined as

$$\frac{\partial}{\partial \mathbf{a}} = \left(\frac{\partial}{\partial a_1}, \frac{\partial}{\partial a_2}, \dots, \frac{\partial}{\partial a_P} \right)^T \quad (8.12)$$

The least mean square error solution, obtained by setting Equation (8.11) to zero, is given by

$$\mathbf{R}_{xx} \mathbf{a} = \mathbf{r}_{xx} \quad (8.13)$$

From Equation (8.13) the predictor coefficient vector is given by

$$\mathbf{a} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xx} \quad (8.14)$$

Equation (8.14) may also be written in an expanded form as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{xx}(1) \\ r_{xx}(2) \\ r_{xx}(3) \\ \vdots \\ r_{xx}(P) \end{pmatrix} \quad (8.15)$$

An alternative formulation of the least square error problem is as follows. For a signal segment of N samples $[x(0), \dots, x(N-1)]$, we can write a set of N linear prediction error equations as

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(-1) & x(-2) & x(-3) & \cdots & x(-P) \\ x(0) & x(-1) & x(-2) & \cdots & x(1-P) \\ x(1) & x(0) & x(-1) & \cdots & x(2-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(N-2) & x(N-3) & x(N-4) & \cdots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{8.16}$$

where $\mathbf{x}^T = [x(-1), \dots, x(-P)]$ is the initial vector. In a compact vector/matrix notation Equation (8.16) can be written as

$$\mathbf{e} = \mathbf{x} - \mathbf{X}\mathbf{a} \tag{8.17}$$

Using Equation (8.17), the sum of squared prediction errors over a block of N samples can be expressed as

$$\mathbf{e}^T \mathbf{e} = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{X}\mathbf{a} + \mathbf{a}^T \mathbf{X}^T \mathbf{X}\mathbf{a} \tag{8.18}$$

The least squared error predictor is obtained by setting the derivative of Equation (8.14) with respect to the parameter vector \mathbf{a} to zero:

$$\frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{a}} = -2\mathbf{x}^T \mathbf{X} + 2\mathbf{a}^T \mathbf{X}^T \mathbf{X} = 0 \tag{8.19}$$

From Equation (8.19), the least square error predictor is given by

$$\mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{x}) \tag{8.20}$$

A comparison of Equations (8.15) and (8.20) shows that in Equation (8.20) the autocorrelation matrix and vector of Equation (8.15) are replaced by the time-averaged estimates as

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1} x(k)x(k-m) \tag{8.21}$$

Equations (8.15) or (8.20) may be solved efficiently by utilising the regular Toeplitz structure of the correlation matrix, \mathbf{R}_{xx} . In a Toeplitz matrix, all the elements on a left–right diagonal are equal. The correlation matrix is also cross-diagonal symmetric. Note that altogether there are only $P + 1$ unique elements $[r_{xx}(0), r_{xx}(1), \dots, r_{xx}(P)]$ in the correlation matrix and the cross-correlation vector. An efficient method for solution of Equation (8.15) or (8.20) is the Levinson–Durbin algorithm, introduced in Section 8.2.2.

8.1.3 EFFECT OF ESTIMATION OF CORRELATION FUNCTION ON LP MODEL SOLUTION

Note that the term $\hat{r}_{xx}(m)$ in Equation (8.21) is only an *estimate* of the correlation function, obtained from a segment of N samples, and as such $\hat{r}_{xx}(m)$ is a random variable with its own mean, variance and probability distribution function. Indeed different segments of an even stationary signal will yield different values of $\hat{r}_{xx}(m)$. The goodness of an estimate depends on the number of samples N used in the estimation of the correlation function and on the signal to noise ratio.

8.1.4 THE INVERSE FILTER: SPECTRAL WHITENING

The all-pole linear predictor model, in Figure 8.5, shapes the spectrum of the input signal by transforming an uncorrelated excitation signal, $u(m)$, to a correlated output signal, $x(m)$. In the frequency domain the input–output relation of the all-pole filter of Figure 8.5 is given by

$$X(f) = \frac{GU(f)}{A(f)} = \frac{E(f)}{1 - \sum_{k=1}^P a_k e^{-j2\pi fk}} \tag{8.22}$$

where $X(f)$, $E(f)$ and $U(f)$ are the spectra of $x(m)$, $e(m)$ and $u(m)$, respectively, G is the input gain factor, and $A(f)$ is the frequency response of the inverse predictor. As the excitation signal $e(m)$ is assumed to have a flat spectrum, it follows that the shape of the signal spectrum, $X(f)$, is due to the frequency response, $1/A(f)$, of the all-pole predictor model. The inverse linear predictor, as the name implies, transforms a correlated signal $x(m)$ back to an uncorrelated flat-spectrum signal $e(m)$. The inverse filter, Figure 8.7, also known as the prediction error filter, is an all-zero finite impulse response filter defined as

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=1}^P a_k x(m-k) \\ &= (\mathbf{a}^{inv})^T \mathbf{x} \end{aligned} \tag{8.23}$$

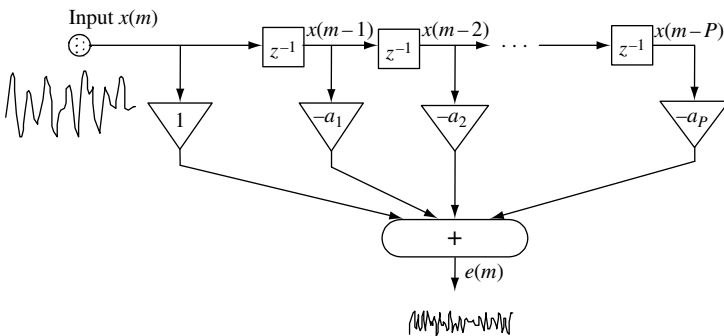


Figure 8.7 Illustration of the inverse (or whitening) filter.

where the inverse filter $(\mathbf{a}^{\text{inv}})^T = [1, -a_1, \dots, -a_P] = [1, -\mathbf{a}]$, and $\mathbf{x}^T = [x(m), \dots, x(m - P)]$. The z -transfer function of the inverse predictor model is given by

$$A(z) = 1 - \sum_{k=1}^P a_k z^{-k} \tag{8.24}$$

A linear predictor model is an all-pole filter, where the poles model the resonance of the signal spectrum. The inverse of an all-pole filter is an all-zero filter, with the zeros situated at the same positions in the pole-zero plot as the poles of the all-pole filter, as illustrated in Figure 8.8. Consequently, the zeros of the inverse filter introduce anti-resonance that cancels out the resonance of the poles of the predictor. The inverse filter has the effect of flattening the spectrum of the input signal, and is also known as a spectral whitening, or de-correlation, filter.

8.1.5 THE PREDICTION ERROR SIGNAL

In general, the prediction error signal is composed of three components:

- (1) the input signal, also called the excitation signal;
- (2) the errors due to the modelling inaccuracies;
- (3) the noise.

The mean square prediction error becomes zero only if the following three conditions are satisfied: (1) the signal is deterministic; (2) the signal is correctly modelled by a predictor of order P ; and (3) the signal is noise-free. For example, a mixture of $P/2$ sine waves can be modelled by a predictor of order P , with zero prediction error. However, in practice, the prediction error is nonzero because information-bearing signals are random, often only approximately modelled by a linear system, and usually observed in noise. The least mean square prediction error, obtained from substitution of Equation (8.13) in Equation (8.10), is

$$E^{(P)} = \mathcal{E}[e^2(m)] = r_{xx}(0) - \sum_{k=1}^P a_k r_{xx}(k) \tag{8.25}$$

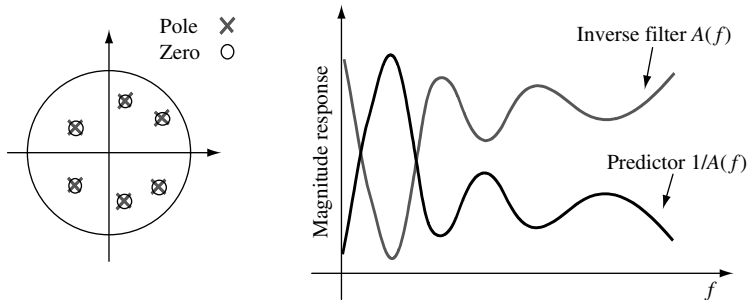


Figure 8.8 Illustration of the pole-zero diagram, and the frequency responses of an all-pole predictor and its all-zero inverse filter.

where $E^{(P)}$ denotes the prediction error for a predictor of order P . The prediction error decreases, initially rapidly and then slowly, with the increasing predictor order up to the correct model order. For the correct model order, the signal $e(m)$ is an uncorrelated zero-mean random process with an autocorrelation function defined as

$$\mathbb{E}[e(m)e(m-k)] = \begin{cases} \sigma_e^2 = G^2 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases} \quad (8.26)$$

where σ_e^2 is the variance of $e(m)$.

Figure 8.9 shows an example of linear prediction analysis of a segment of speech.

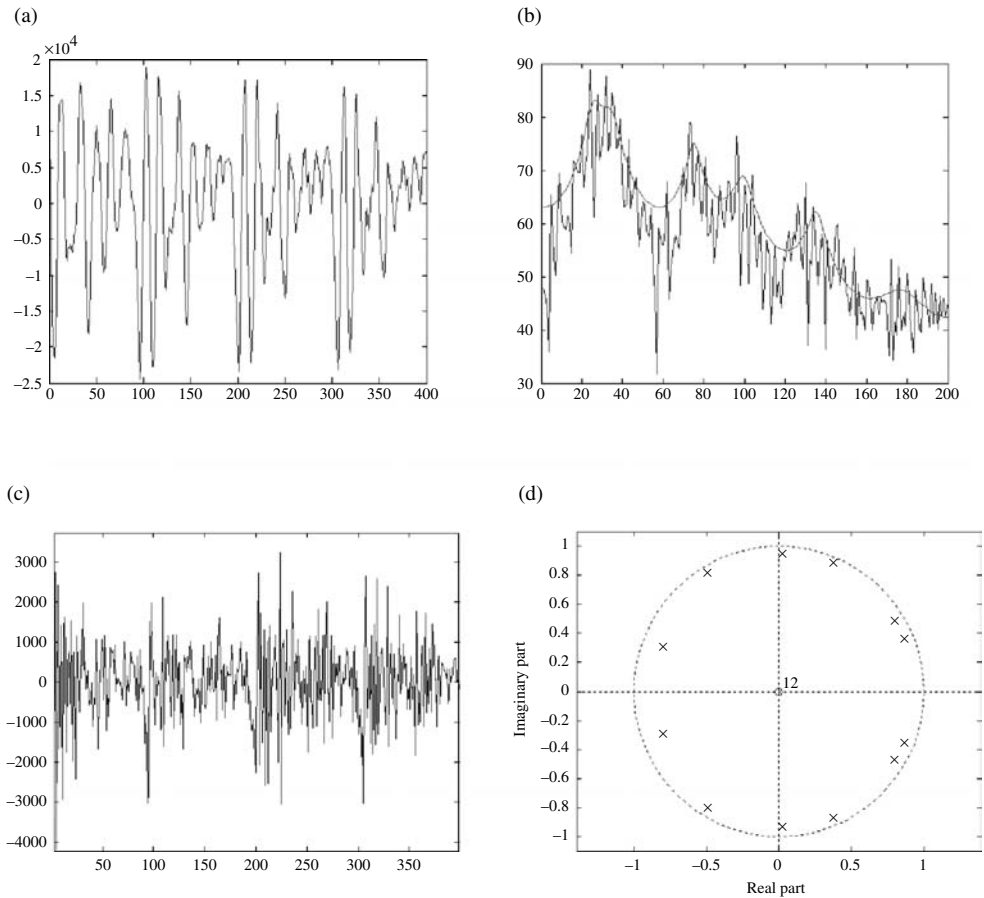


Figure 8.9 (a) A speech segment; (b) FFT spectrum and LP frequency response of speech superimposed; (c) inverse filter output; (d) the poles of LP model, prediction order $P = 12$. Speech sampling rate = 16 kHz.

8.2 FORWARD, BACKWARD AND LATTICE PREDICTORS

The forward predictor model of Equation (8.1) predicts a sample $x(m)$ from a linear combination of P past samples $x(m-1), x(m-2), \dots, x(m-P)$. Similarly, as shown in Figure 8.10, we can define a backward predictor, that predicts a sample $x(m-P)$ from P future samples $x(m-P+1), \dots, x(m)$ as

$$\hat{x}(m-P) = \sum_{k=1}^P c_k x(m-k+1) \quad (8.27)$$

The backward prediction error is defined as the difference between the actual sample and its predicted value:

$$\begin{aligned} b(m) &= x(m-P) - \hat{x}(m-P) \\ &= x(m-P) - \sum_{k=1}^P c_k x(m-k+1) \end{aligned} \quad (8.28)$$

From Equation (8.28), a signal generated by a backward predictor is given by

$$x(m-P) = \sum_{k=1}^P c_k x(m-k+1) + b(m) \quad (8.29)$$

The coefficients of the least square error backward predictor, obtained using a similar method to that of the forward predictor in Section 8.1.1, are given by

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_P \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \quad (8.30)$$

Note that the main difference between Equations (8.30) and (8.15) is that the correlation vector on the right-hand side of the backward predictor, Equation (8.30), is upside-down

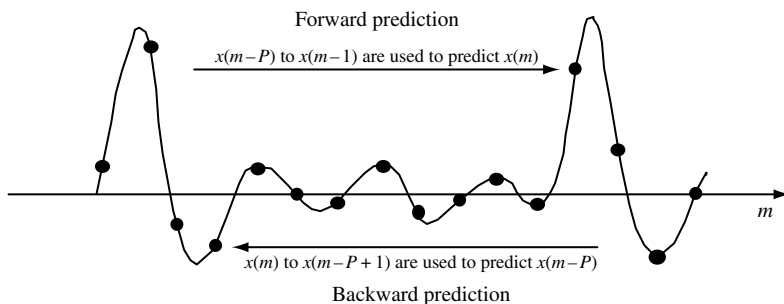


Figure 8.10 Illustration of forward and backward predictors.

compared with the forward predictor, Equation (8.15). Since the correlation matrix is Toeplitz and symmetric, Equation (8.15) for the forward predictor may be rearranged and rewritten in the following form:

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_p \\ a_{p-1} \\ a_{p-2} \\ \vdots \\ a_1 \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \quad (8.31)$$

Comparison of Equations (8.31) and (8.30) shows that the coefficients of the backward predictor are the time-reversed version of the forward predictor

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_p \end{pmatrix} = \begin{pmatrix} a_p \\ a_{p-1} \\ a_{p-2} \\ \vdots \\ a_1 \end{pmatrix} = \mathbf{a}^B \quad (8.32)$$

where the vector \mathbf{a}^B is the reversed version of the vector \mathbf{a} . The relation between the backward and forward predictors is employed in the Levinson–Durbin algorithm to derive an efficient method for calculation of the predictor coefficients, as described in Section 8.2.2.

8.2.1 AUGMENTED EQUATIONS FOR FORWARD AND BACKWARD PREDICTORS

The inverse forward predictor coefficient vector is $[1, -a_1, \dots, -a_p] = [1, -\mathbf{a}^T]$. Equations (8.15) and (8.26) may be combined to yield a matrix equation for the inverse forward predictor coefficients:

$$\begin{pmatrix} r(0) & \mathbf{r}_{xx}^T \\ \mathbf{r}_{xx} & \mathbf{R}_{xx} \end{pmatrix} \begin{pmatrix} 1 \\ -\mathbf{a} \end{pmatrix} = \begin{pmatrix} E^{(P)} \\ \mathbf{0} \end{pmatrix} \quad (8.33)$$

Equation (8.33) is called the ‘augmented forward predictor equation’. Similarly, for the inverse backward predictor, we can define an augmented backward predictor equation as

$$\begin{pmatrix} \mathbf{R}_{xx} & \mathbf{r}_{xx}^B \\ \mathbf{r}_{xx}^{BT} & r(0) \end{pmatrix} \begin{pmatrix} -\mathbf{a}^B \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ E^{(P)} \end{pmatrix} \quad (8.34)$$

where $\mathbf{r}_{xx}^T = [r_{xx}(1), \dots, r_{xx}(P)]$ and $\mathbf{r}_{xx}^{BT} = [r_{xx}(P), \dots, r_{xx}(1)]$. Note that the superscript BT denotes backward and transposed. The augmented forward and backward matrix

Equations (8.33) and (8.34) are used to derive an order-update solution for the linear predictor coefficients as follows.

8.2.2 LEVINSON–DURBIN RECURSIVE SOLUTION

The Levinson–Durbin algorithm was developed by N. Levinson in 1947 and modified by J. Durbin in 1959. The Levinson–Durbin algorithm is a recursive order-update method for calculation of linear predictor coefficients. A forward-prediction error filter of order i can be described in terms of the forward and backward prediction error filters of order $i-1$ as

$$\begin{pmatrix} 1 \\ -a_1^{(i)} \\ \vdots \\ -a_{i-1}^{(i)} \\ -a_i^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -a_1^{(i-1)} \\ \vdots \\ -a_{i-1}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -a_{i-1}^{(i-1)} \\ \vdots \\ -a_1^{(i-1)} \\ 1 \end{pmatrix} \quad (8.35)$$

or in a more compact vector notation as

$$\begin{pmatrix} 1 \\ -\mathbf{a}^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -\mathbf{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -\mathbf{a}^{(i-1)\text{B}} \\ 1 \end{pmatrix} \quad (8.36)$$

where k_i is the reflection coefficient. The proof of Equation (8.36) and the derivation of the value of the reflection coefficient for k_i follow shortly. Similarly, a backward prediction error filter of order i is described in terms of the forward and backward prediction error filters of order $i-1$ as

$$\begin{pmatrix} -\mathbf{a}^{(i-1)\text{B}} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\mathbf{a}^{(i-1)\text{B}} \\ 1 \end{pmatrix} + k_i \begin{pmatrix} 1 \\ -\mathbf{a}^{(i-1)} \\ 0 \end{pmatrix} \quad (8.37)$$

To prove the order-update Equation (8.36), [or alternatively Equation (8.37)], we multiply both sides of the equation by the $(i+1) \times (i+1)$ augmented matrix $\mathbf{R}_{xx}^{(i+1)}$ and use the equality

$$\mathbf{R}_{xx}^{(i+1)} = \begin{pmatrix} \mathbf{R}_{xx}^{(i)} & \mathbf{r}_{xx}^{(i)\text{B}} \\ \mathbf{r}_{xx}^{(i)\text{BT}} & r_{xx}(0) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{xx}(0) \mathbf{r}_{xx}^{(i)\text{T}} \\ \mathbf{r}_{xx}^{(i)} & \mathbf{R}_{xx}^{(i)} \end{pmatrix} \quad (8.38)$$

to obtain

$$\begin{pmatrix} \mathbf{R}_{xx}^{(i)} & \mathbf{r}_{xx}^{(i)\text{B}} \\ \mathbf{r}_{xx}^{(i)\text{BT}} & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\mathbf{a}^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_{xx}^{(i)} & \mathbf{r}_{xx}^{(i)\text{B}} \\ \mathbf{r}_{xx}^{(i)\text{BT}} & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\mathbf{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} \mathbf{r}_{xx}(0) \mathbf{r}_{xx}^{(i)\text{T}} \\ \mathbf{r}_{xx}^{(i)} & \mathbf{R}_{xx}^{(i)} \end{pmatrix} \begin{pmatrix} 0 \\ -\mathbf{a}^{(i-1)\text{B}} \\ 1 \end{pmatrix} \quad (8.39)$$

where in Equations (8.38) and (8.39) $\mathbf{r}_{xx}^{(i)T} = [r_{xx}(1), \dots, r_{xx}(i)]$, and $\mathbf{r}_{xx}^{(i)BT} = [r_{xx}(i), \dots, r_{xx}(1)]$ is the reversed version of $\mathbf{r}_{xx}^{(i)T}$. Matrix–vector multiplication of both sides of Equation (8.39) and the use of Equations (8.33) and (8.34) yields

$$\begin{pmatrix} E^{(i)} \\ \mathbf{0}^{(i)} \end{pmatrix} = \begin{pmatrix} E^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ \Delta^{(i-1)} \end{pmatrix} + k_i \begin{pmatrix} \Delta^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ E^{(i-1)} \end{pmatrix} \quad (8.40)$$

where

$$\begin{aligned} \Delta^{(i-1)} &= [1 \quad -\mathbf{a}^{(i-1)}]^T \mathbf{r}_{xx}^{(i)B} \\ &= r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \end{aligned} \quad (8.41)$$

If Equation (8.40) is true, it follows that Equation (8.36) must also be true. The conditions for Equation (8.40) to be true are

$$E^{(i)} = E^{(i-1)} + k_i \Delta^{(i-1)} \quad (8.42)$$

and

$$0 = \Delta^{(i-1)} + k_i E^{(i-1)} \quad (8.43)$$

From Equation (8.43),

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \quad (8.44)$$

Substitution of $\Delta^{(i-1)}$ from Equation (8.44) into Equation (8.42) yields

$$\begin{aligned} E^{(i)} &= E^{(i-1)}(1 - k_i^2) \\ &= E^{(0)} \prod_{j=1}^i (1 - k_j^2) \end{aligned} \quad (8.45)$$

Note that it can be shown that $\Delta^{(i)}$ is the cross-correlation of the forward and backward prediction errors:

$$\Delta^{(i-1)} = \mathcal{E}[b^{(i-1)}(m-1)e^{(i-1)}(m)] \quad (8.46)$$

The parameter $\Delta^{(i-1)}$ is known as the partial correlation.

Levinson–Durbin Algorithm

The Durbin algorithm starts with a predictor of order zero for which $E^{(0)} = r_{xx}(0)$. The algorithm then computes the coefficients of a predictor of order i , using the coefficients of a

predictor of order $i - 1$. In the process of solving for the coefficients of a predictor of order P , the solutions for the predictor coefficients of all orders less than P are also obtained:

$$E^{(0)} = r_{xx}(0) \tag{8.47}$$

For $i = 1, \dots, P$

$$\Delta^{(i-1)} = r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \tag{8.48}$$

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \tag{8.49}$$

$$a_i^{(i)} = -k_i \tag{8.50}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad 1 \leq j \leq i-1 \tag{8.51}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{8.52}$$

8.2.3 LATTICE PREDICTORS

The lattice structure, shown in Figure 8.11, is a cascade connection of similar units, with each unit specified by a single parameter k_i , known as the *reflection coefficient*. A major attraction of a lattice structure is its modular form and the relative ease with which the model order can

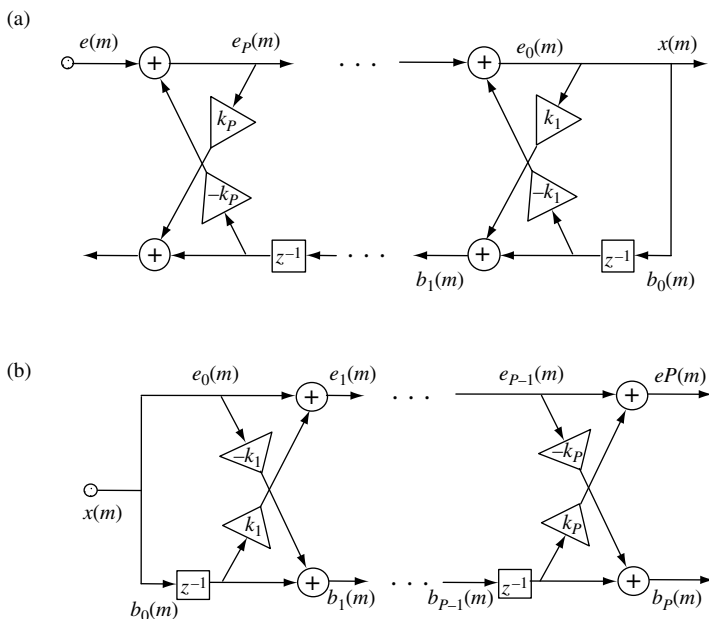


Figure 8.11 (a) A lattice predictor and (b) the inverse lattice predictor.

be extended. A further advantage is that, for a stable model, the magnitude of k_i is bounded by unity ($|k_i| < 1$), and therefore it is relatively easy to check a lattice structure for stability. The lattice structure is derived from the forward and backward prediction errors as follows. An order-update recursive equation can be obtained for the forward prediction error by multiplying both sides of Equation (8.32) by the input vector $[x(m), x(m-1), \dots, x(m-i)]$:

$$e^{(i)}(m) = e^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \quad (8.53)$$

Similarly, we can obtain an order-update recursive equation for the backward prediction error by multiplying both sides of Equation (8.37) by the input vector $[x(m-i), x(m-i+1), \dots, x(m)]$ as

$$b^{(i)}(m) = b^{(i-1)}(m-1) - k_i e^{(i-1)}(m) \quad (8.54)$$

Equations (8.53) and (8.54) are interrelated and may be implemented by a lattice network, as shown in Figure 8.11. Minimisation of the squared forward prediction error of Equation (8.53) over N samples yields

$$k_i = \frac{\sum_{m=0}^{N-1} e^{(i-1)}(m) b^{(i-1)}(m-1)}{\sum_{m=0}^{N-1} [e^{(i-1)}(m)]^2} \quad (8.55)$$

Note that a similar relation for k_i can be obtained through minimisation of the squared backward prediction error of Equation (8.54) over N samples. The reflection coefficients are also known as the normalised partial correlation (PARCOR) coefficients.

8.2.4 ALTERNATIVE FORMULATIONS OF LEAST SQUARE ERROR PREDICTION

The methods described above for derivation of the predictor coefficients are based on minimisation of either the forward or the backward prediction error. In this section, we consider alternative methods based on the minimisation of the sum of the forward and backward prediction errors.

Burg's Method

Burg's method is based on minimisation of the sum of the forward and backward squared prediction errors. The squared error function is defined as

$$E_{\text{fb}}^{(i)} = \sum_{m=0}^{N-1} \left\{ [e^{(i)}(m)]^2 + [b^{(i)}(m)]^2 \right\} \quad (8.56)$$

Substitution of Equations (8.53) and (8.54) in Equation (8.56) yields

$$E_{\text{fb}}^{(i)} = \sum_{m=0}^{N-1} \left\{ [e^{(i-1)}(m) - k_i b^{(i-1)}(m-1)]^2 + [b^{(i-1)}(m-1) - k_i e^{(i-1)}(m)]^2 \right\} \quad (8.57)$$

Minimisation of $E_{\text{fb}}^{(i)}$ with respect to the reflection coefficients k_i yields

$$k_i = \frac{2 \sum_{m=0}^{N-1} e^{(i-1)}(m)b^{(i-1)}(m-1)}{\sum_{m=0}^{N-1} \left\{ [e^{(i-1)}(m)]^2 + [b^{(i-1)}(m-1)]^2 \right\}} \quad (8.58)$$

Simultaneous Minimisation of the Backward and Forward Prediction Errors

From Equation (8.32), the backward predictor coefficient vector is the reversed version of the forward predictor coefficient vector. Hence a predictor of order P can be obtained through simultaneous minimisation of the sum of the squared backward and forward prediction errors defined by the following equation:

$$\begin{aligned} E_{\text{fb}}^{(P)} &= \sum_{m=0}^{N-1} \left\{ [e^{(P)}(m)]^2 + [b^{(P)}(m)]^2 \right\} \\ &= \sum_{m=0}^{N-1} \left\{ \left[x(m) - \sum_{k=1}^P a_k x(m-k) \right]^2 + \left[x(m-P) - \sum_{k=1}^P a_k x(m-P+k) \right]^2 \right\} \\ &= (\mathbf{x} - \mathbf{X}\mathbf{a})^T (\mathbf{x} - \mathbf{X}\mathbf{a}) + (\mathbf{x}^{\text{B}} - \mathbf{X}^{\text{B}}\mathbf{a})^T (\mathbf{x}^{\text{B}} - \mathbf{X}^{\text{B}}\mathbf{a}) \end{aligned} \quad (8.59)$$

where \mathbf{X} and \mathbf{x} are the signal matrix and vector defined by Equations (8.16) and (8.17), and similarly \mathbf{X}^{B} and \mathbf{x}^{B} are the signal matrix and vector for the backward predictor. Using an approach similar to that used in derivation of Equation (8.20), the minimisation of the mean squared error function of Equation (8.59) yields

$$\mathbf{a} = (\mathbf{X}^T \mathbf{X} + \mathbf{X}^{\text{BT}} \mathbf{X}^{\text{B}})^{-1} (\mathbf{X}^T \mathbf{x} + \mathbf{X}^{\text{BT}} \mathbf{x}^{\text{B}}) \quad (8.60)$$

Note that, for an ergodic signal, as the signal length N increases Equation (8.60) converges to the so-called 'normal' Equation (8.14).

8.2.5 PREDICTOR MODEL ORDER SELECTION

One procedure for the determination of the correct model order is to increment the model order, and monitor the differential change in the error power, until the change levels off. The incremental change in error power with the increasing model order from $i-1$ to i is defined as

$$\Delta E^{(i)} = E^{(i-1)} - E^{(i)} \quad (8.61)$$

Figure 8.12 illustrates the decrease in the normalised mean square prediction error with the increasing predictor length for a speech signal. The order P beyond which the decrease in the error power $\Delta E^{(P)}$ becomes less than a threshold is taken as the model order.

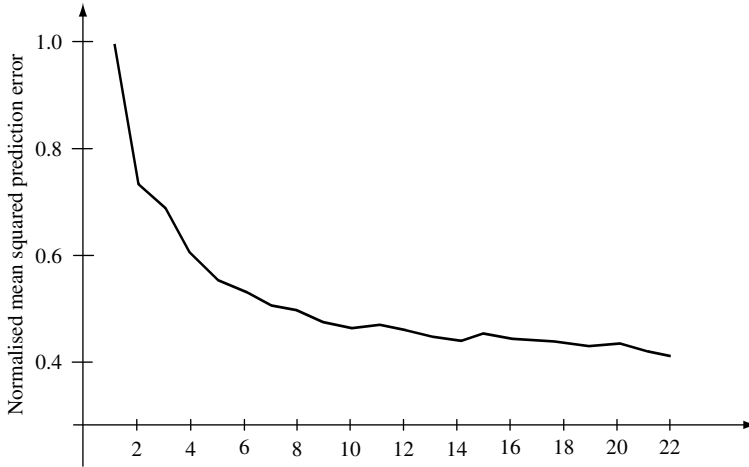


Figure 8.12 Illustration of the decrease in the normalised mean squared prediction error with the increasing predictor length for a speech signal.

In linear prediction two coefficients are required for modelling each spectral peak of the signal spectrum, for example, the modelling of a signal with K dominant resonance in the spectrum needs $P = 2K$ coefficients. Hence a procedure for model selection is to examine the power spectrum of the signal process, and to set the model order to twice the number of significant spectral peaks in the spectrum.

When the model order is less than the correct order, the signal is undermodelled. In this case the prediction error is not well de-correlated and will be more than the optimal minimum. A further consequence of undermodelling is a decrease in the spectral resolution of the model: adjacent spectral peaks of the signal could be merged and appear as a single spectral peak when the model order is too small. When the model order is larger than the correct order, the signal is overmodelled. An overmodelled problem can result in an ill-conditioned matrix equation, unreliable numerical solutions and the appearance of spurious spectral peaks in the model.

8.3 SHORT- AND LONG-TERM PREDICTORS

For quasiperiodic signals, such as voiced speech, there are two types of correlation structures that can be utilised for a more accurate prediction. These are:

- (1) the short-term correlation, which is the correlation of each sample with the P immediate past samples $-x(m-1), \dots, x(m-P)$;
- (2) the long-term correlation, which is the correlation of a sample $x(m)$ with, say, $2Q+1$ similar samples a pitch period T away $-x(m-T+Q), \dots, x(m-T-Q)$.

Figure 8.13 is an illustration of the short-term relation of a sample with the P immediate past samples and its long-term relation with the samples a pitch period away. The short-term

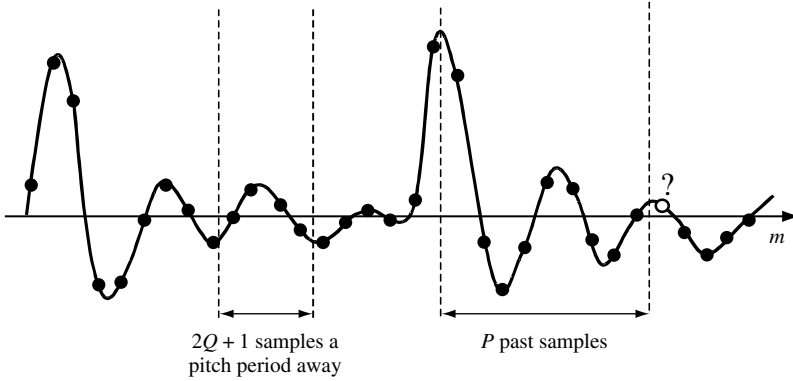


Figure 8.13 Illustration of the short-term relation of a sample with the P immediate past samples and the long-term relation with the samples a pitch period away.

correlation of a signal may be modelled by the linear prediction equation [Equation (8.3)]. The remaining correlation, in the prediction error signal $e(m)$, is called the ‘long-term correlation’. The long-term correlation in the prediction error signal may be modelled by a pitch predictor, defined as

$$\hat{e}(m) = \sum_{k=-Q}^Q p_k e(m - T - k) \tag{8.62}$$

where p_k are the coefficients of a long-term predictor of order $2Q + 1$. The pitch period T can be obtained from the autocorrelation function of $x(m)$ or that of $e(m)$: it is the first nonzero time-lag where the autocorrelation function attains a maximum. Assuming that the long-term correlation is correctly modelled, the prediction error of the long-term filter is a completely random signal with a white spectrum, and is given by

$$\begin{aligned} \varepsilon(m) &= e(m) - \hat{e}(m) \\ &= e(m) - \sum_{k=-Q}^Q p_k e(m - T - k) \end{aligned} \tag{8.63}$$

Minimisation of $\mathcal{E}[e^2(m)]$ results in the following solution for the pitch predictor:

$$\begin{pmatrix} p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{Q-1} \\ p_Q \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(2Q) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(2Q-1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(2Q-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(2Q) & r_{xx}(2Q-1) & r_{xx}(2Q-2) & \cdots & r_{xx}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{xx}(T-Q) \\ r_{xx}(T-Q+1) \\ \vdots \\ r_{xx}(T+Q-1) \\ r_{xx}(T+Q) \end{pmatrix} \tag{8.64}$$

An alternative to the separate, cascade, modelling of the short- and long-term correlations is to combine the short- and long-term predictors into a single model described as

$$x(m) = \underbrace{\sum_{k=1}^P a_k x(m-k)}_{\text{short-term prediction}} + \underbrace{\sum_{k=-Q}^Q p_k x(m-k-T)}_{\text{long-term prediction}} + \varepsilon(m) \tag{8.65}$$

In Equation (8.65), each sample is expressed as a linear combination of P immediate past samples and $2Q + 1$ samples a pitch period away. Minimisation of $\mathcal{E}[e^2(m)]$ results in the following solution for the pitch predictor:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \\ p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{+Q} \end{pmatrix} = \begin{pmatrix} r(0) & r(1) & \dots & r(P-1) & r(T+Q-1) & r(T+Q) & \dots & r(T-Q-1) \\ r(1) & r(0) & \dots & r(P-2) & r(T+Q-2) & r(T+Q-1) & \dots & r(T+Q-2) \\ r(2) & r(1) & \dots & r(P-3) & r(T+Q-3) & r(T+Q-2) & \dots & r(T+Q-3) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \dots & r(0) & r(T+Q-P) & r(T+Q-P+1) & \dots & r(T+Q-P) \\ r(T+Q-1) & r(T+Q-2) & \dots & r(T+Q-P) & r(0) & r(1) & \dots & r(2Q) \\ r(T+Q) & r(T+Q-1) & \dots & r(T+Q-P+1) & r(1) & r(0) & \dots & r(2Q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(T-Q-1) & r(T-Q-2) & \dots & r(T-Q-P) & r(2Q) & r(2Q-1) & \dots & r(0) \end{pmatrix}^{-1} \times \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T+Q) \\ r(T+Q-1) \\ \vdots \\ r(T-Q) \end{pmatrix} \tag{8.66}$$

In Equation (8.66), for simplicity the subscript xx of $r_{xx}(k)$ has been omitted. In Chapter 10, the predictor model of Equation (8.65) is used for interpolation of a sequence of missing samples

8.4 MAP ESTIMATION OF PREDICTOR COEFFICIENTS

The posterior probability density function of a predictor coefficient vector \mathbf{a} , given a signal \mathbf{x} and the initial samples \mathbf{x}_1 , can be expressed, using Bayes's rule, as

$$f_{\mathbf{A}|\mathbf{X}, \mathbf{x}_1}(\mathbf{a}|\mathbf{x}, \mathbf{x}_1) = \frac{f_{\mathbf{X}|\mathbf{A}, \mathbf{x}_1}(\mathbf{x}|\mathbf{a}, \mathbf{x}_1) f_{\mathbf{A}|\mathbf{x}_1}(\mathbf{a}|\mathbf{x}_1)}{f_{\mathbf{X}|\mathbf{x}_1}(\mathbf{x}|\mathbf{x}_1)} \tag{8.67}$$

In Equation (8.67), the pdfs are conditioned on P initial signal samples, $\mathbf{x}_1 = [x(-P), x(-P+1), \dots, x(-1)]$. Note that, for a given set of samples $[\mathbf{x}, \mathbf{x}_1]$, $f_{\mathbf{X}|\mathbf{x}_1}(\mathbf{x}|\mathbf{x}_1)$ is a constant, and it is reasonable to assume that $f_{\mathbf{A}|\mathbf{x}_1}(\mathbf{a}|\mathbf{x}_1) = f_{\mathbf{A}}(\mathbf{a})$.

8.4.1 PROBABILITY DENSITY FUNCTION OF PREDICTOR OUTPUT

The pdf $f_{X|A, X_1}(x|a, x_1)$ of the signal x , given the predictor coefficient vector a and the initial samples x_1 , is equal to the pdf of the input signal e :

$$f_{X|A, X_1}(x|a, x_1) = f_E(x - Xa) \tag{8.68}$$

where the input signal vector is given by

$$e = -Xa \tag{8.69}$$

and $f_E(e)$ is the pdf of e . Equation (8.64) can be expanded as

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(-1) & x(-2) & x(-3) & \cdots & x(-P) \\ x(0) & x(-1) & x(-2) & \cdots & x(1-P) \\ x(1) & x(0) & x(-1) & \cdots & x(2-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(N-2) & x(N-3) & x(N-4) & \cdots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{8.70}$$

Assuming that the input excitation signal $e(m)$ is a zero-mean, uncorrelated, Gaussian process with a variance of σ_e^2 , the likelihood function in Equation (8.68) becomes

$$\begin{aligned} f_{X|A, X_1}(x|a, x_1) &= f_E(x - Xa) \\ &= \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp \left[\frac{1}{2\sigma_e^2} (x - Xa)^T (x - Xa) \right] \end{aligned} \tag{8.71}$$

An alternative form of Equation (8.71) can be obtained by rewriting Equation (8.70) in the following form:

$$\begin{pmatrix} e_0 \\ e_1 \\ e_3 \\ e_4 \\ \vdots \\ e_{N-1} \end{pmatrix} = \begin{pmatrix} -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 \end{pmatrix} \begin{pmatrix} x_{-P} \\ x_{-P+1} \\ x_{-P+2} \\ x_{-P+3} \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{8.72}$$

In a compact notation Equation (8.72) can be written as

$$e = Ax \tag{8.73}$$

Using Equation (8.73), and assuming that the excitation signal, $e(m)$, is a zero-mean, uncorrelated process with variance σ_e^2 , the likelihood function of Equation (8.71) can be written as

$$f_{X|A, X_1}(x|a, x_1) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp \left(-\frac{1}{2\sigma_e^2} x^T A^T A x \right) \tag{8.74}$$

8.4.2 USING THE PRIOR PDF OF THE PREDICTOR COEFFICIENTS

The prior pdf of the predictor coefficient vector is assumed to have a Gaussian distribution with a mean vector μ_a and a covariance matrix Σ_{aa} :

$$f_A(\mathbf{a}) = \frac{1}{(2\pi)^{P/2} |\Sigma_{aa}|^{1/2}} \exp \left[-\frac{1}{2} (\mathbf{a} - \mu_a)^T \Sigma_{aa}^{-1} (\mathbf{a} - \mu_a) \right] \quad (8.75)$$

Substituting Equations (8.72) and (8.76) in Equation (8.67), the posterior pdf of the predictor coefficient vector $f_{A|X, X_I}(\mathbf{a}|\mathbf{x}, \mathbf{x}_I)$ can be expressed as

$$f_{A|X, X_I}(\mathbf{a}|\mathbf{x}, \mathbf{x}_I) = \frac{1}{f_{X|X_I}(\mathbf{x}|\mathbf{x}_I)} \frac{1}{(2\pi)^{(N+P)/2} \sigma_e^N |\Sigma_{aa}|^{1/2}} \times \exp \left\{ -\frac{1}{2} \left[\frac{1}{\sigma_e^2} (\mathbf{x} - \mathbf{X}\mathbf{a})^T (\mathbf{x} - \mathbf{X}\mathbf{a}) + (\mathbf{a} - \mu_a)^T \Sigma_{aa}^{-1} (\mathbf{a} - \mu_a) \right] \right\} \quad (8.76)$$

The maximum *a posteriori* estimate is obtained by maximising the log-likelihood function:

$$\frac{\partial}{\partial \mathbf{a}} \left[\ln f_{A|X, X_I}(\mathbf{a}|\mathbf{x}, \mathbf{x}_I) \right] = \frac{\partial}{\partial \mathbf{a}} \left[\frac{1}{\sigma_e^2} (\mathbf{x} - \mathbf{X}\mathbf{a})^T (\mathbf{x} - \mathbf{X}\mathbf{a}) + (\mathbf{a} - \mu_a)^T \Sigma_{aa}^{-1} (\mathbf{a} - \mu_a) \right] = 0 \quad (8.77)$$

This yields

$$\hat{\mathbf{a}}^{\text{MAP}} = (\Sigma_{aa} \mathbf{X}^T \mathbf{X} + \sigma_e^2 \mathbf{I})^{-1} \Sigma_{aa} \mathbf{X}^T \mathbf{x} + \sigma_e^2 (\Sigma_{aa} \mathbf{X}^T \mathbf{X} + \sigma_e^2 \mathbf{I})^{-1} \mu_a \quad (8.78)$$

Note that, as the Gaussian prior tends to a uniform prior, the determinant covariance matrix Σ_{aa} of the Gaussian prior increases, and the MAP solution tends to the least square error solution:

$$\hat{\mathbf{a}}^{\text{LS}} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{x}) \quad (8.79)$$

Similarly, as the observation length N increases, the signal matrix term $\mathbf{X}^T \mathbf{X}$ becomes more significant than Σ_{aa} and again the MAP solution tends to a least square error solution.

8.5 FORMANT-TRACKING LP MODELS

Formants are the resonance frequencies of speech. In the application of linear prediction to speech, the poles of linear prediction model the resonance at formants of speech. The z -transfer function of the linear prediction model of speech can be expressed in a cascade form as

$$X(z) = G(z)V(z)L(z) \quad (8.80)$$

where $G(z)$, $V(z)$ and $L(z)$ are the z -transforms of glottal pulse, vocal tract and lip radiation. The vocal tract can be modelled by formant-tracking LP models, which may be expressed as

$$V(z, m) = G(m) \prod_{k=1}^N \frac{1}{1 - 2r_k \varphi_k(m) z^{-1} + r_k^2(m) z^{-2}} \quad (8.81)$$

where $\varphi_k(m)$, $r_k(m)$ and $G(m)$ are the time-varying angular frequency and radii of the poles and the gain of a second-order section of the linear prediction model. There is not a one-to-one relationship between poles of linear prediction models and formants. In fact, depending on the model order, the linear prediction model may associate more than one pole with a formant or, conversely, more than one formant with a pole. The poles of the linear prediction model are the formant candidates – the raw data from which the formants and their models are estimated.

The spectral resonance at formant can be characterized by a parameter vector comprising the frequency (F_k), bandwidth (B_k) and magnitude of the resonance (M_k), and their temporal slope of variation as

$$F_k = [F_k, B_k, M_k, \Delta F_k, \Delta B_k, \Delta M_k] \quad k = 1, \dots, M \tag{8.82}$$

where Δ denotes the slope of the trajectory of a feature vector over time, e.g. $\Delta F_k(t)$ for the k th formant at frame t is obtained as

$$\Delta F_k(t) = \frac{\sum_{m=1}^N m [F_k(t+m) - F_k(t-m)]}{\sum_{m=1}^N 2m^2} \quad k = 1, \dots, M \tag{8.83}$$

There are three issues in modelling formants: (1) the modelling of the distribution of formants using a probability model; (2) tracking of the trajectory of each formant using a classifier; and (3) smoothing the trajectory of formants. Formant tracking methods are the subject of current research.

Using the formant tracks a formant-tracking LP model can be constructed. Formant tracking models provide a framework for modelling the inter-correlation of LP models across successive frames speech. Formant-tracking LP models can be used for speech synthesis and for speech enhancement through de-noising the parameters of LP model of speech.

Figure 8.14 illustrates the spectrogram of the frequency response of linear prediction model of a segment of speech with the estimates of the formant tracks superimposed.

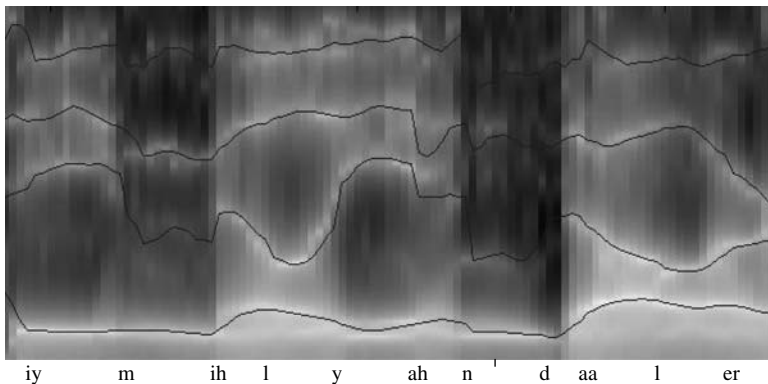


Figure 8.14 An example of formant tracks superimposed on an LP spectrogram. Light areas on the spectrogram correspond to high energy.

8.6 SUB-BAND LINEAR PREDICTION MODEL

The poles of a linear predictor equation model the signal spectrum over its full bandwidth. The distribution of the poles of the LP model over the signal spectrum depends on the signal correlation and spectral structure. Generally, the poles redistribute themselves over the spectrum to minimize the mean square prediction error criterion. An alternative to a conventional LP model is to divide the signal into a number of sub-bands and to model the signal within each band with a linear prediction model, as shown in Figure 8.15.

The advantages of using a sub-band LP model are as follows:

- (1) sub-band linear prediction allows the designer to allocate different numbers of parameters to different bands;
- (2) the solution of a full-band linear predictor Equation (8.10) or (8.16) requires the inversion of a relatively large correlation matrix, whereas the solution of the sub-band LP models requires the inversion of a number of smaller correlation matrices with better numerical stability properties, e.g. a predictor of order 18 requires the inversion of an 18×18 matrix, whereas three sub-band predictors of order 6 require the inversion of three 6×6 matrices;
- (3) sub-band linear prediction is useful for applications such as noise reduction where a sub-band approach can offer more flexibility and better performance.

In sub-band linear prediction, the signal $x(m)$ is passed through a bank of N band-pass filters, and is split into N sub-band signals, $x_k(m)$, $k = 1, \dots, N$. The k th sub-band signal is modelled using a low-order linear prediction model as

$$x_k(m) = \sum_{i=1}^{P_k} a_k(i)x_k(m-i) + g_k e_k(m) \quad (8.84)$$

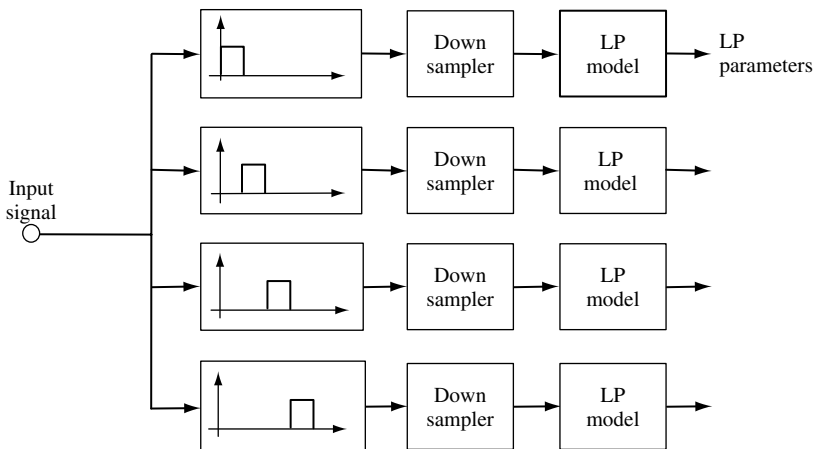


Figure 8.15 Configuration of a sub-band linear prediction model.

where $[a_k, g_k]$ are the coefficients and the gain of the predictor model for the k th sub-band. The choice of the model order P_k depends on the width of the sub-band and on the signal correlation structure within each sub-band. The power spectrum of the input excitation of an ideal LP model for the k th sub-band signal can be expressed as

$$P_{EE}(f, k) = \begin{cases} 1 & f_{k,\text{start}} < f < f_{k,\text{end}} \\ 0 & \text{otherwise} \end{cases} \quad (8.85)$$

where $f_{k,\text{start}}, f_{k,\text{end}}$ are the start and end frequencies of the k th sub-band signal. The autocorrelation function of the excitation function in each sub-band is a sinc function given by

$$r_{ee}(m) = B_k \text{sinc}[m(B_k - f_{k0})/2] \quad (8.86)$$

where B_k and f_{k0} are the bandwidth and the centre frequency of the k th sub-band, respectively. To ensure that each sub-band LP parameters only model the signal within that sub-band, the sub-band signals are down-sampled, as shown in Figure 8.15.

Note that a problem with sub-band linear prediction is that the pole frequencies may happen at or near the cut-off frequency of sub-bands. To avoid this problem formant trajectory estimation described in the previous section may be used to track the frequencies of the poles of the signal. The centre frequency of the sub-bands follows the formant tracks.

8.7 SIGNAL RESTORATION USING LINEAR PREDICTION MODELS

Linear prediction models are extensively used in speech and audio signal restoration. For a noisy signal, linear prediction analysis models the combined spectra of the signal and the noise processes. For example, the frequency spectrum of a linear prediction model of speech, observed in additive white noise, would be flatter than the spectrum of the noise-free speech, owing to the influence of the flat spectrum of white noise. In this section we consider the estimation of the coefficients of a predictor model from noisy observations, and the use of linear prediction models in signal restoration. The noisy signal $y(m)$ is modelled as

$$\begin{aligned} y(m) &= x(m) + n(m) \\ &= \sum_{k=1}^P a_k x(m-k) + e(m) + n(m) \end{aligned} \quad (8.87)$$

where the signal $x(m)$ is modelled by a linear prediction model with coefficients a_k and random input $e(m)$, and it is assumed that the noise $n(m)$ is additive. The least square error predictor model of the noisy signal $y(m)$ is given by

$$\mathbf{R}_{yy} \hat{\mathbf{a}} = \mathbf{r}_{yy} \quad (8.88)$$

where \mathbf{R}_{yy} and \mathbf{r}_{yy} are the autocorrelation matrix and vector of the noisy signal $y(m)$. For an additive noise model, Equation (8.88) can be written as

$$(\mathbf{R}_{xx} + \mathbf{R}_{nn})(\mathbf{a} + \tilde{\mathbf{a}}) = (\mathbf{r}_{xx} + \mathbf{r}_{nn}) \quad (8.89)$$

where $\tilde{\mathbf{a}}$ is the error in the predictor coefficients vector due to the noise. A simple method for removing the effects of noise is to subtract an estimate of the autocorrelation of the noise from that of the noisy signal. The drawback of this approach is that, owing to random variations of noise, correlation subtraction can cause numerical instability in Equation (8.88) and result in spurious solutions. In the following, we formulate the pdf of the noisy signal and describe an iterative signal-restoration/parameter-estimation procedure developed by Lee and Oppenheim.

From Bayes's rule, the MAP estimate of the predictor coefficient vector \mathbf{a} , given an observation signal vector, $\mathbf{y} = [y(0), y(1), \dots, y(N-1)]$, and the initial samples vector \mathbf{x}_1 , is

$$f_{A|Y, X_1}(\mathbf{a}|\mathbf{y}, \mathbf{x}_1) = \frac{f_{Y|A, X_1}(\mathbf{y}|\mathbf{a}, \mathbf{x}_1)f_{A, X_1}(\mathbf{a}, \mathbf{x}_1)}{f_{Y, X_1}(\mathbf{y}, \mathbf{x}_1)} \quad (8.90)$$

Consider the variance of the signal \mathbf{y} in the argument of the term $f_{Y|A, X_1}(\mathbf{y}|\mathbf{a}, \mathbf{x}_1)$ in Equation (8.90). The innovation (i.e. prediction error) of $y(m)$ can be defined as

$$\begin{aligned} \varepsilon(m) &= y(m) - \sum_{k=1}^P \hat{a}_k y(m-k) \\ &= e(m) + \varepsilon(m) + n(m) - \sum_{k=1}^P \hat{a}_k n(m-k) \end{aligned} \quad (8.91)$$

The variance of $y(m)$, given the previous P samples and the coefficient vector $\hat{\mathbf{a}}$; is the variance of the innovation signal given by

$$\text{Var}[y(m)|y(m-1), \dots, y(m-P), \hat{\mathbf{a}}] = \sigma_e^2 + \sigma_\varepsilon^2 + \sigma_n^2 - \sigma_n^2 \sum_{k=1}^P \hat{a}_k^2 \quad (8.92)$$

where σ_e^2 , σ_ε^2 and σ_n^2 are the variance of the excitation signal, the error in innovation due to noise and the noise, respectively. From Equation (8.92), the variance of $[y(m)|y(m-1), \dots, y(m-P), \hat{\mathbf{a}}]$ is a function of the coefficient vector $\hat{\mathbf{a}}$. Consequently, maximisation of $f_{Y|A, X_1}(\mathbf{y}|\hat{\mathbf{a}}, \mathbf{x}_1)$ with respect to the vector $\hat{\mathbf{a}}$ is a nonlinear and nontrivial exercise.

Lim and Oppenheim proposed the following iterative process where an estimate $\hat{\mathbf{a}}$ of the predictor coefficient vector is used to make an estimate $\hat{\mathbf{x}}$ of the signal vector, and the signal estimate $\hat{\mathbf{x}}$ is then used to improve the estimate of the parameter vector $\hat{\mathbf{a}}$, and the process is iterated until convergence. The posterior pdf of the noise-free signal \mathbf{x} , given the noisy signal \mathbf{y} and an estimate of the parameter vector $\hat{\mathbf{a}}$, is given by

$$f_{X|A, Y}(\mathbf{x}|\hat{\mathbf{a}}, \mathbf{y}) = \frac{f_{Y|A, X}(\mathbf{y}|\hat{\mathbf{a}}, \mathbf{x})f_{X|A}(\mathbf{x}|\hat{\mathbf{a}})}{f_{Y|A}(\mathbf{y}|\hat{\mathbf{a}})} \quad (8.93)$$

Consider the likelihood term $f_{Y|A, X}(\mathbf{y}|\hat{\mathbf{a}}, \mathbf{x})$. Since the noise is additive, we have

$$\begin{aligned} f_{Y|A, X}(\mathbf{y}|\hat{\mathbf{a}}, \mathbf{x}) &= f_N(\mathbf{y} - \mathbf{x}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left[-\frac{1}{2\sigma_n^2}(\mathbf{y} - \mathbf{x})^T(\mathbf{y} - \mathbf{x})\right] \end{aligned} \quad (8.94)$$

Assuming that the input of the predictor model is a zero-mean Gaussian process with variance σ_e^2 , the pdf of the signal \mathbf{x} given an estimate of the predictor coefficient vector \mathbf{a} is

$$\begin{aligned} f_{Y|A, X}(\mathbf{x}|\hat{\mathbf{a}}) &= \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{e}^T \mathbf{e}\right) \\ &= \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \hat{\mathbf{A}}^T \hat{\mathbf{A}} \mathbf{x}\right) \end{aligned} \quad (8.95)$$

where $\mathbf{e} = \hat{\mathbf{A}}\mathbf{x}$ as in Equation (8.73). Substitution of Equations (8.94) and (8.95) in Equation (8.93) yields

$$f_{X|A, Y}(\mathbf{x}|\hat{\mathbf{a}}, \mathbf{y}) = \frac{1}{f_{Y|A}(\mathbf{y}|\hat{\mathbf{a}})} \frac{1}{(2\pi\sigma_n\sigma_e)^N} \exp\left[-\frac{1}{2\sigma_n^2} (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) - \frac{1}{2\sigma_e^2} \mathbf{x}^T \hat{\mathbf{A}}^T \hat{\mathbf{A}} \mathbf{x}\right] \quad (8.96)$$

In Equation (8.92), for a given signal \mathbf{y} and coefficient vector $\hat{\mathbf{a}}$, $f_{Y|A}(\mathbf{y}|\hat{\mathbf{a}})$ is a constant. From Equation (8.92), the ML signal estimate is obtained by maximising the log-likelihood function as

$$\frac{\partial}{\partial \mathbf{a}} [\ln f_{X|A, Y}(\mathbf{x}|\hat{\mathbf{a}}, \mathbf{y})] = \frac{\partial}{\partial \mathbf{x}} \left[-\frac{1}{2\sigma_e^2} \mathbf{x}^T \hat{\mathbf{A}}^T \hat{\mathbf{A}} \mathbf{x} - \frac{1}{2\sigma_n^2} (\mathbf{y} - \mathbf{x})^T (\mathbf{y} - \mathbf{x}) \right] = 0 \quad (8.97)$$

which gives

$$\hat{\mathbf{x}} = \sigma_e^2 \left(\sigma_n^2 \hat{\mathbf{A}}^T \hat{\mathbf{A}} + \sigma_e^2 \mathbf{I} \right)^{-1} \mathbf{y} \quad (8.98)$$

The signal estimate of Equation (8.98) can be used to obtain an updated estimate of the predictor parameter. Assuming that the signal is a zero-mean Gaussian process, the estimate of the predictor parameter vector \mathbf{a} is given by

$$\hat{\mathbf{a}}(\hat{\mathbf{x}}) = (\hat{\mathbf{x}}^T \hat{\mathbf{x}})^{-1} (\hat{\mathbf{x}}^T \hat{\mathbf{y}}) \quad (8.99)$$

Equations (8.98) and (8.99) form the basis for an iterative signal restoration/parameter estimation method.

8.7.1 FREQUENCY-DOMAIN SIGNAL RESTORATION USING PREDICTION MODELS

The following algorithm is a frequency-domain implementation of the linear prediction model-based restoration of a signal observed in additive white noise.

Initialisation: set the initial signal estimate to noisy signal $\hat{\mathbf{x}}_0 = \mathbf{y}$,

For iterations $i = 0, 1, \dots$

Step 1: estimate the predictor parameter vector $\hat{\mathbf{a}}_i$:

$$\hat{\mathbf{a}}_i(\hat{\mathbf{x}}_i) = (\hat{\mathbf{x}}_i^T \hat{\mathbf{x}}_i)^{-1} (\hat{\mathbf{x}}_i^T \hat{\mathbf{y}}_i) \quad (8.100)$$

Step 2: calculate an estimate of the model gain G using Parseval's theorem:

$$\frac{1}{N} \sum_{f=0}^{N-1} \frac{\hat{G}^2}{\left| 1 - \sum_{k=1}^P \hat{a}_{k,i} e^{-j2\pi fk/N} \right|^2} = \sum_{m=0}^{N-1} y^2(m) - N\hat{\sigma}_n^2 \quad (8.101)$$

where $\hat{a}_{k,i}$ are the coefficient estimates at iteration i , and $N\hat{\sigma}_n^2$ is the energy of white noise over N samples.

Step 3: calculate an estimate of the power spectrum of speech model,

$$\hat{P}_{X_i X_i}(f) = \frac{\hat{G}^2}{\left| 1 - \sum_{k=1}^P \hat{a}_{k,i} e^{-j2\pi fk/N} \right|^2} \quad (8.102)$$

Step 4: calculate the Wiener filter frequency response,

$$\hat{W}_i(f) = \frac{\hat{P}_{X_i X_i}(f)}{\hat{P}_{X_i X_i}(f) + \hat{P}_{N_i N_i}(f)} \quad (8.103)$$

where $\hat{P}_{N_i N_i}(f) = \hat{\sigma}_n^2$ is an estimate of the noise power spectrum.

Step 5: filter the magnitude spectrum of the noisy speech as

$$\hat{X}_{i+1}(f) = \hat{W}_i(f)Y(f) \quad (8.104)$$

Restore the time domain signal \hat{x}_{i+1} by combining $\hat{X}_{i+1}(f)$ with the phase of noisy signal and the complex signal-to-time domain.

Step 6: goto step 1 and repeat until convergence, or for a specified number of iterations.

Figure 8.16 illustrates a block diagram configuration of a Wiener filter using a linear prediction estimate of the signal spectrum. Figure 8.17 illustrates the result of an iterative restoration of the spectrum of a noisy speech signal.

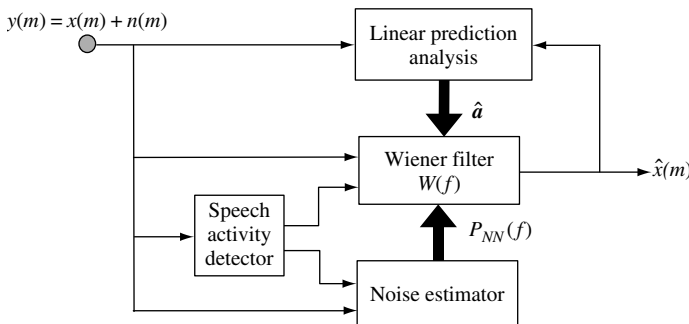


Figure 8.16 Iterative signal restoration based on linear prediction model of speech.

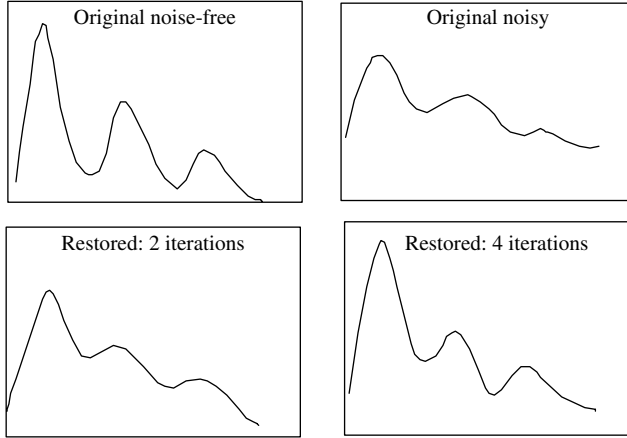


Figure 8.17 Illustration of restoration of a noisy signal with iterative linear prediction-based method.

8.7.2 IMPLEMENTATION OF SUB-BAND LINEAR PREDICTION WIENER FILTERS

Assuming that the noise is additive, the noisy signal in each sub-band is modelled as

$$y_k(m) = x_k(m) + n_k(m) \quad (8.105)$$

The Wiener filter in the frequency domain can be expressed in terms of the power spectra, or in terms of LP model frequency responses, of the signal and noise process as

$$\begin{aligned} W_k(f) &= \frac{P_{X,k}(f)}{P_{Y,k}(f)} \\ &= \frac{g_{X,k}^2 |A_{Y,k}(f)|^2}{|A_{X,k}(f)|^2 g_{Y,k}^2} \end{aligned} \quad (8.106)$$

where $P_{X,k}(f)$ and $P_{Y,k}(f)$ are the power spectra of the clean signal and the noisy signal for the k th subband, respectively. From Equation (8.112) the square-root Wiener filter is given by

$$W_k^{1/2}(f) = \frac{g_{X,k} |A_{Y,k}(f)|}{|A_{X,k}(f)| g_{Y,k}} \quad (8.107)$$

The linear prediction Wiener filter of Equation (8.107) can be implemented in the time domain with a cascade of a linear predictor of the clean signal, followed by an inverse predictor filter of the noisy signal as expressed by the following relations (see Figure 8.18):

$$z_k(m) = \sum_{i=1}^P a_{Xk}(i) z_k(m-i) + \frac{g_X}{g_Y} y_k(m) \quad (8.108)$$

$$\hat{x}_k(m) = \sum_{i=0}^P a_{Yk}(i) z_k(m-i) \quad (8.109)$$

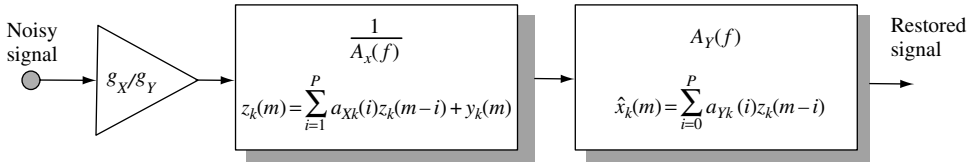


Figure 8.18 A cascade implementation of the LP square-root Wiener filter.

where $\hat{x}_k(m)$ is the restored estimate of the clear speed signal $x_k(m)$, $z_k(m)$ is an intermediate signal, $a_{y_k}(i)$ are the coefficients of the linear predictor model of the noisy signal for the k th sub-band and $a_{x_k}(i)$ is an estimate of the coefficients of the linear prediction model of the k th sub-band of clean speech.

8.8 SUMMARY

Linear prediction models are used in a wide range of signal processing applications from low-bit-rate speech/video coding to model-based spectral analysis. We began this chapter with an introduction to linear prediction theory, and considered different methods of formulation of the prediction problem and derivations of the predictor coefficients. The main attraction of the linear prediction method is the closed-form solution of the predictor coefficients, and the availability of a number of efficient and relatively robust methods for solving the prediction equation, such as the Levinson–Durbin method.

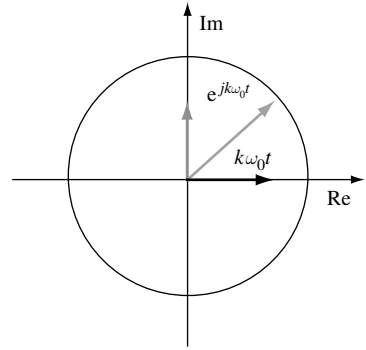
In Section 8.2, we considered the forward, backward and lattice predictors. Although the direct-form implementation of the linear predictor is the most convenient method, for many applications, such as transmission of the predictor coefficients in speech coding, it is advantageous to use the lattice form of the predictor. This is because the lattice form can be conveniently checked for stability, and furthermore a perturbation of the parameter of any section of the lattice structure has a limited and more localised effect. In Section 8.3, we considered a modified form of linear prediction that models the short-term and long-term correlations of the signal. This method can be used for the modelling of signals with a quasiperiodic structure such as voiced speech. In Section 8.4, we considered MAP estimation and the use of a prior pdf for derivation of the predictor coefficients. In Section 8.5, the sub-band linear prediction method was formulated. Finally in Section 8.6, a linear prediction model was applied to the restoration of a signal observed in additive noise.

BIBLIOGRAPHY

- Akaike H. (1970) Statistical predictor identification. *Ann. Inst. Statist. Math.*, **22**: 203–218.
 Akaike H. (1974) A new look at statistical model identification. *IEEE Trans. Autom. Control*, **AC-19**: 716–723.
 Anderson O.D. (1976) *Time Series Analysis and Forecasting, The Box–Jenkins Approach*. Butterworth, London.
 Ayre A.J. (1972) *Probability and Evidence*. Columbia University Press, New York.
 Box G.E.P and Jenkins G.M. (1976) *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA.
 Burg J.P. (1975) Maximum entropy spectral analysis. Ph.D. thesis, Stanford University, Stanford, CA.
 Cohen J. and Cohen, P. (1975). *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Halsted, New York.

- Draper N.R. and Smith, H. (1981). *Applied Regression Analysis* (2nd edn). Wiley, New York.
- Durbin J. (1959) Efficient estimation of parameters in moving average models. *Biometrika*, **46**: 306–318.
- Durbin J. (1960) The fitting of time series models. *Rev. Int. Statist. Inst.*, **28**: 233–244.
- Fuller W.A (1976) *Introduction to Statistical Time Series*. Wiley, New York.
- Hansen J.H. and Clements M.A. (1987). Iterative speech enhancement with spectral constraints. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-87*, Dallas, Tx, April, Vol. 1, pp. 189–192.
- Hansen J.H. and Clements M.A. (1988). Constrained iterative speech enhancement with application to automatic speech recognition. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-88*, New York, April, vol. 1, pp. 561–564.
- Hocking R.R. (1996) *The Analysis of Linear Models*. Wiley, New York.
- Kobatake H., Inari J. and Kakuta S. (1978) Linear prediction coding of speech signals in a high ambient noise environment. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing*, April, pp. 472–475.
- Lim J.S. and Oppenheim A.V. (1978) All-pole modelling of degraded speech. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-26**(3): 197–210.
- Lim J.S. and Oppenheim A.V. (1979) Enhancement and bandwidth compression of noisy speech. *Proc. IEEE*, **67**: 1586–1604.
- Makoul J.(1975) Linear prediction: a tutorial review. *Proc. IEEE*, **63**: 561–580.
- Markel J.D. and Gray A.H. (1976) *Linear Prediction of Speech*. Springer, New York.
- Rabiner L.R. and Schafer R.W. (1976) *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ.
- Tong H. (1975) Autoregressive model fitting with noisy data by Akaike's information criterion. *IEEE Trans. Inform. Theory*, **IT-23**: 409–448.
- Stockham T.G. and Cannon T.M. and Ingebreetsen R.B (1975) Blind deconvolution through digital signal processing. *IEEE Proc.*, **63**(4): 678–692.

9



Power Spectrum and Correlation

9.1 Power Spectrum and Correlation	242
9.2 Fourier Series: Representation of Periodic Signals	243
9.3 Fourier Transform: Representation of Aperiodic Signals	245
9.4 Nonparametric Power Spectrum Estimation	249
9.5 Model-based Power Spectrum Estimation	254
9.6 High-resolution Spectral Estimation Based on Subspace Eigenanalysis	259
9.7 Summary	265
Bibliography	266

The power spectrum reveals the existence, or the absence, of repetitive patterns and correlation structures in a signal process. These structural patterns are important in a wide range of applications such as data forecasting, signal coding, signal detection, radar, pattern recognition and decision-making systems. The most common method of spectral estimation is based on the fast Fourier transform (FFT). For many applications, FFT-based methods produce sufficiently good results. However, more advanced methods of spectral estimation can offer better frequency resolution, and less variance.

This chapter begins with an introduction to the Fourier series and transform and the basic principles of spectral estimation. The classical methods for power spectrum estimation are based on periodograms. Various methods of averaging periodograms, and their effects on the variance of spectral estimates, are considered. We then study the maximum entropy and the model-based spectral estimation methods. We also consider several high-resolution spectral estimation methods, based on eigenanalysis, for the estimation of sinusoids observed in additive white noise.

9.1 POWER SPECTRUM AND CORRELATION

The power spectrum and correlation functions are Fourier transform pairs and hence they contain exactly the same information presented in different domains: the correlation is a function of time domain and the power spectrum is a function of frequency domain. The correlation and the power spectrum are widely used in communication signal processing systems in such applications as system analysis, feature extraction, modelling, signal detection and decoders.

The power spectrum of a signal gives the distribution of the signal power among various frequencies and shows the existence, and also the relative power, of repetitive patterns and/or random structures in a signal.

Correlation is a measure of self-similarity of a signal with its delayed version. Like the power spectrum, the correlation function reveals information on the periodic or random structure of the signal.

The strength of the Fourier transform in signal analysis and pattern recognition is its ability to reveal spectral structures that may be used to characterize a signal. This is illustrated in Figure 9.1 for the two extreme cases of a sine wave and a purely random signal. For a periodic signal, the power is concentrated in extremely narrow bands of frequencies, indicating the existence of structure and the predictable character of the signal. In the case of a pure sine wave, as shown in Figure 9.1(a), the signal power is concentrated in one frequency. For a purely random signal, as shown in Figure 9.1(b), the signal power is spread equally in the frequency domain, indicating the lack of structure in the signal.

In general, the more correlated or predictable a signal is, the more concentrated its power spectrum and, conversely, the more random or unpredictable a signal, the wider the spread of its power spectrum. Therefore the power spectrum of a signal can be used to deduce the existence of repetitive structures or correlated patterns in the signal process. Such information is crucial in detection, decision-making and estimation problems, and in systems analysis.

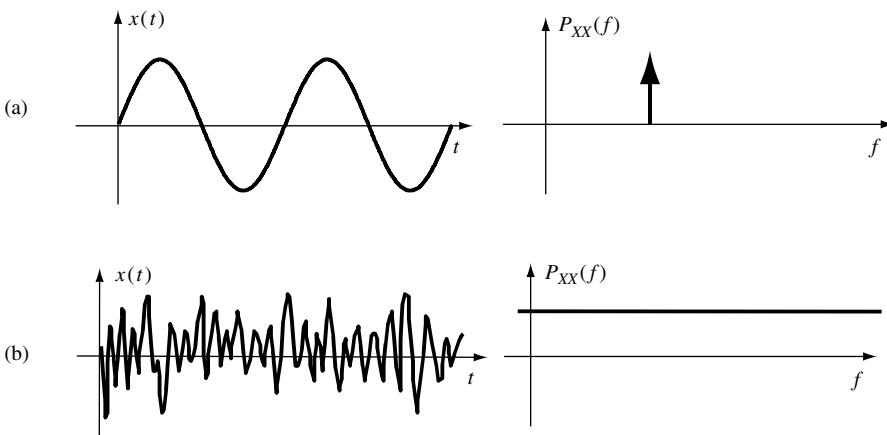


Figure 9.1 The concentration/spread of power in frequency indicates the correlated or random character of a signal: (a) a predictable signal, (b) a random signal.

9.2 FOURIER SERIES: REPRESENTATION OF PERIODIC SIGNALS

The following three sinusoidal functions form the *basis functions* for the Fourier analysis:

$$x_1(t) = \cos \omega_0 t \quad (9.1)$$

$$x_2(t) = \sin \omega_0 t \quad (9.2)$$

$$x_3(t) = \cos \omega_0 t + j \sin \omega_0 t = e^{j\omega_0 t} \quad (9.3)$$

where $\omega_0 = 2\pi f_0$. A cosine function is an even function about time $t = 0$ and a sine function is an odd function. A weighted combination of a sine and cosine at angular frequency ω_0 can model any phase of a sinusoidal signal component of $x(t)$ at that frequency. Figure 9.2(a) shows the sine and cosine components of the complex exponential (cisoidal) signal of Equation (9.3), and Figure 9.2(b) shows a vector representation of the complex exponential in a complex plane with real (Re) and imaginary (Im) dimensions. The Fourier basis functions are periodic with an angular frequency of ω_0 rad/s and a period of $T_0 = 2\pi/\omega_0 = 1/F_0$ s, where F_0 is the frequency in Hz. The following properties make the sinusoids an ideal choice as the elementary building block basis functions for signal analysis and synthesis:

- (1) Orthogonality – two sinusoidal functions of *different* frequencies have the following orthogonal property:

$$\int_{-\infty}^{\infty} \sin(\omega_1 t) \sin(\omega_2 t) dt = -\frac{1}{2} \int_{-\infty}^{\infty} \cos(\omega_1 + \omega_2)t dt + \frac{1}{2} \int_{-\infty}^{\infty} \cos(\omega_1 - \omega_2)t dt = 0 \quad (9.4)$$

For sinusoids the integration interval can be taken over one period [i.e. $T = 2\pi/(\omega_1 + \omega_2)$ and $T = 2\pi/(\omega_1 - \omega_2)$]. Similar equations can be derived for the product of cosines, or sine and cosine, of different frequencies. Orthogonality implies that the sinusoidal basis

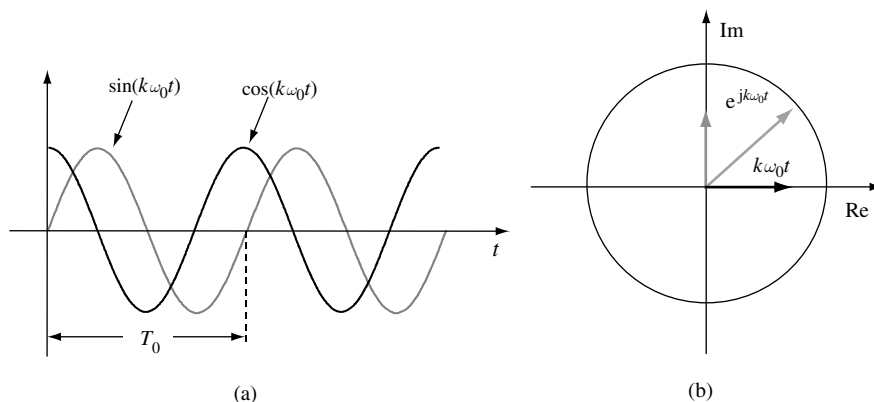


Figure 9.2 Fourier basis functions: (a) real and imaginary signals of a complex sinusoid; (b) vector representation of a complex exponential.

functions are independent and can be processed independently. For example, in a graphic equaliser we can change the relative amplitudes of one set of frequencies, such as the bass, without affecting other frequencies, and in music coding the signals in different frequency bands are coded independently and allocated different numbers of bits.

- (2) The sine and cosine components of $e^{-j\omega_0 t}$ have only a relative phase difference of $\pi/2$ or equivalently a relative time delay of a quarter of one period, i.e. $T_0/4$. This allows decomposition of a signal in terms of orthogonal cosine and sine components.
- (3) Sinusoidal functions are infinitely differentiable. This is a useful property, as most signal analysis, synthesis and processing methods require the signals to be differentiable.
- (4) A useful consequence of transforms, such as the Fourier and the Laplace transforms is that differential analysis on the time domain signal becomes simple algebraic operations on the transformed signal.

Associated with the complex exponential function $e^{j\omega_0 t}$ is a set of harmonically related complex exponentials of the form

$$[1, e^{\pm j\omega_0 t}, e^{\pm j2\omega_0 t}, e^{\pm j3\omega_0 t}, \dots] \quad (9.5)$$

The set of exponential signals in Equation (9.5) is periodic with a fundamental frequency $\omega_0 = 2\pi/T_0 = 2\pi F_0$ where T_0 is the period and F_0 is the fundamental frequency. These signals form the set of *basis functions* for the Fourier analysis. Any linear combination of these signals of the form

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \quad (9.6)$$

is also a periodic signal with a period of T_0 . Conversely any periodic signal $x(t)$ can be synthesized from a linear combination of harmonically related exponentials.

The Fourier series representation of a periodic signal is given by the following synthesis and analysis equations:

Analysis equation

$$c_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-j2\pi k F_0 t} dt \quad k = \dots -1, 0, 1, \dots \quad (9.7)$$

Synthesis equation

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t} \quad k = \dots -1, 0, 1, \dots \quad (9.8)$$

The complex-valued coefficient c_k conveys the amplitude (a measure of the strength) and the phase of the frequency content of the signal at $k\omega_0$ Hz. Note from the analysis, Equation (9.7), that the coefficient c_k may be interpreted as a measure of the correlation of the signal $x(t)$ and the complex exponential $e^{-jk\omega_0 t}$.

The representation of a signal in the form of Equation (9.7) as the sum of its constituent harmonics is referred to as the ‘complex Fourier series’ representation. The set of complex coefficients $\dots c_{-1}, c_0, c_1, \dots$ is known as the frequency spectrum of the signal.

Equation (9.7) can be used as a synthesizer (as in a music synthesizers) to generate a signal as a weighted combination of its elementary frequencies. Note from Equations (9.7) and (9.8) that the complex exponentials that form a periodic signal occur only at discrete frequencies which are integer-multiple harmonics of the fundamental frequency F_0 . Therefore the spectrum of a periodic signal, with a period of T_0 , is *discrete* in frequency with *discrete spectral lines* spaced at integer multiples of $F_0 = 1/T_0$.

9.3 FOURIER TRANSFORM: REPRESENTATION OF APERIODIC SIGNALS

The Fourier representation of nonperiodic signals can be obtained by considering a nonperiodic signal as a special case of a periodic signal with an infinite period. If the period of a signal is infinite, then the signal does not repeat itself and is nonperiodic.

The Fourier series representation of periodic signals consists of harmonically related sinusoidal signals with discrete spectra, where the spectral lines are spaced at integer multiples of the fundamental frequency. Now consider the discrete spectra of a periodic signal with a period of T_0 , as shown in Figure 9.3(a). As the period T_0 increases, the fundamental frequency $F_0 = 1/T_0$ decreases, and successive spectral lines become more closely spaced. In the limit, as the period tends to infinity (i.e. as the signal becomes nonperiodic) the discrete spectral lines merge and form a continuous spectrum [Figure 9.3(b)]. Therefore, the Fourier equations for a nonperiodic signal (known as the Fourier transform) must reflect the fact that the frequency spectrum of a non-periodic signal is continuous. Hence, to obtain the Fourier transform relations the discrete-frequency variables and the discrete summation

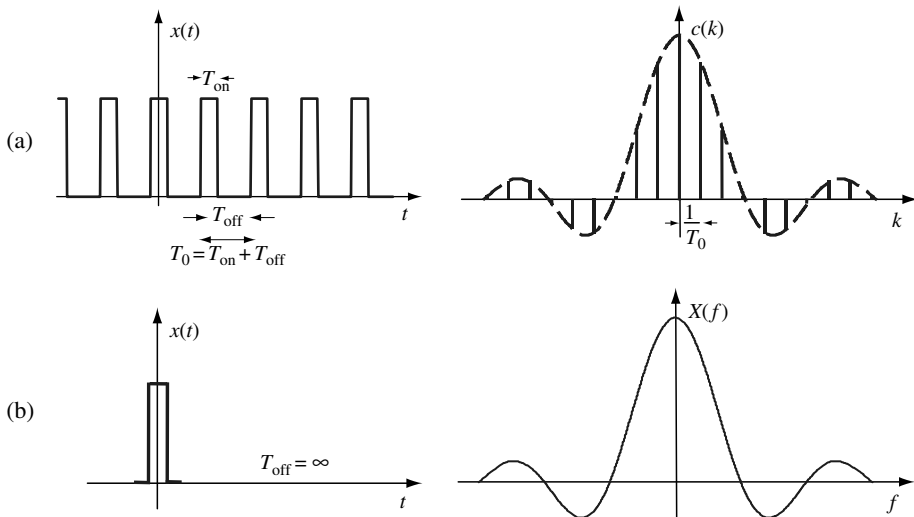


Figure 9.3 (a) A periodic pulse train and its line spectrum. (b) A single pulse from the periodic train in (a) with an imagined ‘off’ duration of infinity; its spectrum is the envelope of the spectrum of the periodic signal in (a).

operations in the Fourier series, Equations (9.7) and (9.8) are replaced by their continuous-frequency counterparts. That is, the discrete summation sign, Σ , is replaced by the continuous summation integral sign, \int , the discrete harmonic of the fundamental frequency, kF_0 , is replaced by the continuous frequency variable, f , and the discrete frequency spectrum, c_k , is replaced by a continuous frequency spectrum, say $X(f)$.

The Fourier synthesis and analysis equations for nonperiodic signals, known as the 'Fourier transform pair', are given by

Fourier transform (analysis) equation

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (9.9)$$

Inverse Fourier transform (synthesis) equation

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft} df \quad (9.10)$$

Note from Equation (9.9) that $X(f)$ may be interpreted as a measure of the correlation of the signal $x(t)$ and the complex sinusoid $e^{-j2\pi ft}$.

The condition for the existence (i.e. computability) of the Fourier transform integral of a signal $x(t)$ is that the signal must have finite energy, that is

$$\text{Signal energy} = \int_{-\infty}^{\infty} |x(t)|^2 dt < \infty \quad (9.11)$$

9.3.1 DISCRETE FOURIER TRANSFORM

Discrete Fourier transform, Figure 9.4, is the Fourier transform adopted for discrete-time signals. DFT is a method of sampling the spectrum of a signal. When a nonperiodic signal is sampled, its Fourier transform becomes a periodic but *continuous* function of frequency, as shown in Equation (9.9). The discrete Fourier transform is derived from sampling the Fourier transform of a discrete-time signal. For a finite-duration discrete-time signal, $x(m)$, of length N samples, the discrete Fourier transform (DFT) is defined as N uniformly spaced spectral samples:

Discrete Fourier transform

$$X(k) = \sum_{m=0}^{N-1} x(m)e^{-j(2\pi/N)mk} \quad k = 0, \dots, N-1 \quad (9.12)$$

Inverse discrete fourier transform

$$x(m) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j(2\pi/N)mk} \quad m = 0, \dots, N-1 \quad (9.13)$$

Note that the basis functions of a DFT are: $1, e^{-j(2\pi/N)}, e^{-j(4\pi/N)}, \dots, e^{-j[(N-1)\pi/N]}$. The DFT equation can be written in the form of a linear system transformation, $\mathbf{X} = \mathbf{W}\mathbf{x}$, as the

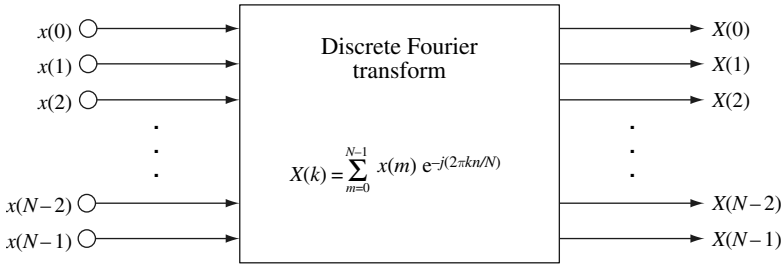


Figure 9.4 Illustration of the DFT as a parallel-input, parallel-output processor.

transformation of an input vector $\mathbf{x} = [x(0), x(1), \dots, x(N-1)]$ to an output vector $\mathbf{X} = [X(0)X(1) \dots X(N-1)]$, or as

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-2) \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & e^{-j(2\pi/N)} & e^{-j(4\pi/N)} & \dots & e^{-j[2(N-2)\pi/N]} & e^{-j[2(N-1)\pi/N]} \\ 1 & e^{-j(4\pi/N)} & e^{-j(8\pi/N)} & \dots & e^{-j[4(N-2)\pi/N]} & e^{-j[4(N-2)\pi/N]} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & e^{-j[2(N-2)\pi/N]} & e^{-j[4(N-2)\pi/N]} & \dots & e^{-j[2(N-2)^2\pi/N]} & e^{-j[2(N-2)(N-1)\pi/N]} \\ 1 & e^{-j[2(N-1)\pi/N]} & e^{-j[4(N-1)\pi/N]} & \dots & e^{-j[2(N-1)(N-2)\pi/N]} & e^{-j[(N-1)^2\pi/N]} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-2) \\ x(N-1) \end{bmatrix} \quad (9.14)$$

where the elements of the transformation matrix are $w_{km} = e^{-j(2\pi/N)km}$. The DFT spectrum consists of N uniformly spaced samples taken from one period (2π) of the continuous spectrum of the discrete time signal $x(m)$. At a sampling rate of F_s the discrete-frequency index k corresponds to kF_s/N Hz.

A periodic signal has a discrete spectrum. Conversely any discrete frequency spectrum belongs to a periodic signal. Hence *the implicit assumption in the DFT theory is that the input signal, $x(m)$, is periodic with a period equal to the observation window length of N samples.* Note that DFT is equivalent to the Fourier series.

9.3.2 TIME/FREQUENCY RESOLUTIONS, THE UNCERTAINTY PRINCIPLE

Signals such as speech, music or image are composed of nonstationary – i.e. time-varying and/or space varying – events. For example, speech is composed of a string of short-duration sounds called phonemes, and an image is composed of various objects. When using the DFT it is desirable to have a high enough time and space resolution in order to obtain the spectral

characteristics of each individual elementary event or object in the input signal. However, there is a fundamental trade-off between the length, i.e. the time or space resolution, of the input signal and the frequency resolution of the output spectrum. The DFT takes as the input a window of N uniformly spaced discrete-time samples $[x(0), x(1), \dots, x(N-1)]$ with a total duration of $\Delta T = N \cdot T_s$, and outputs N spectral samples $[X(0), X(1), \dots, X(N-1)]$ spaced uniformly between 0 Hz and the sampling frequency $F_s = 1/T_s$ Hz. Hence the frequency resolution of the DFT spectrum Δf , i.e. the frequency space between successive frequency samples, is given by

$$\Delta f = \frac{1}{\Delta T} = \frac{1}{NT_s} = \frac{F_s}{N} \quad (9.15)$$

Note that the frequency resolution, Δf , and the time resolution, ΔT , are inversely proportional in that they cannot both be simultaneously increased, in fact $\Delta T \Delta f = 1$. This is known as the uncertainty principle.

9.3.3 ENERGY-SPECTRAL DENSITY AND POWER-SPECTRAL DENSITY

Energy, or power, spectrum analysis is concerned with the distribution of the signal energy or power in the frequency domain. For a deterministic discrete-time signal, the energy-spectral density is defined as

$$E(f) = |X(f)|^2 = \left| \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi f m} \right|^2 \quad (9.16)$$

The energy spectrum of $x(m)$ may be expressed as the Fourier transform of the autocorrelation function of $x(m)$:

$$\begin{aligned} E(f) &= |X(f)|^2 = X(f)X^*(f) \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi f m} \end{aligned} \quad (9.17)$$

where the variable $r_{xx}(m)$ is the autocorrelation function of $x(m)$.

Theoretically, the Fourier transform exists only for finite-energy signals. An important theoretical class of signals is that of stationary stochastic signals, which, as a consequence of the stationarity condition, are infinitely long and have infinite energy, and therefore do not possess a Fourier transform. For stochastic signals, the quantity of interest is the power-spectral density, defined as the Fourier transform of the autocorrelation function:

$$P_{XX}(f) = \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi f m} \quad (9.18)$$

where the autocorrelation function $r_{xx}(m)$ is defined as

$$r_{xx}(m) = \mathcal{E}[x(m)x(m+k)] \quad (9.19)$$

In practice, the autocorrelation function, a measure of self-similarity of a signal with its delayed version, is estimated from a signal record of length N samples as

$$\hat{r}_{xx}(m) = \frac{1}{N - |m|} \sum_{k=0}^{N-|m|-1} x(k)x(k+m), \quad k = 0, \dots, N-1 \quad (9.20)$$

As the correlation lag m approaches the record length N , the estimate of $\hat{r}_{xx}(m)$ is obtained from the average of fewer samples and has a higher variance. A triangular window may be used to 'down-weight' the correlation estimates for larger values of lag m . The triangular window has the form

$$w(m) = \begin{cases} 1 - \frac{|m|}{N}, & |m| \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (9.21)$$

Multiplication of Equation (9.20) by the window of Equation (9.21) yields

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-|m|-1} x(k)x(k+m) \quad (9.22)$$

The expectation of the windowed correlation estimate $\hat{r}_{xx}(m)$ is given by

$$\begin{aligned} \mathcal{E}[\hat{r}_{xx}(m)] &= \frac{1}{N} \sum_{k=0}^{N-|m|-1} \mathcal{E}[x(k)x(k+m)] \\ &= \left(1 - \frac{|m|}{N}\right) r_{xx}(m) \end{aligned} \quad (9.23)$$

In Jenkins and Watts (1968), it is shown that the variance of $\hat{r}_{xx}(m)$ is given by

$$\text{Var}[\hat{r}_{xx}(m)] \approx \frac{1}{N} \sum_{k=-\infty}^{\infty} [r_{xx}^2(k) + r_{xx}(k-m)r_{xx}(k+m)] \quad (9.24)$$

From Equations (9.23) and (9.24), $\hat{r}_{xx}(m)$ is an asymptotically unbiased and consistent estimate.

9.4 NONPARAMETRIC POWER SPECTRUM ESTIMATION

The classic method for estimation of the power spectral density of an N -sample record is the periodogram introduced by Sir Arthur Schuster in 1899. The periodogram is defined as

$$\begin{aligned} \hat{P}_{XX}(f) &= \frac{1}{N} \left| \sum_{m=0}^{N-1} x(m)e^{-j2\pi fm} \right|^2 \\ &= \frac{1}{N} |X(f)|^2 \end{aligned} \quad (9.25)$$

The power-spectral density function, or power spectrum for short, defined in Equation (9.25) is the basis of nonparametric methods of spectral estimation. Owing to the finite length and the random nature of most signals, the spectra obtained from different records of a signal vary randomly about an average spectrum. A number of methods have been developed to reduce the variance of the periodogram.

9.4.1 THE MEAN AND VARIANCE OF PERIODOGRAMS

The mean of the periodogram is obtained by taking the expectation of Equation (9.25):

$$\begin{aligned}
 \mathcal{E}[\hat{P}_{XX}(f)] &= \frac{1}{N} \mathcal{E}[|X(f)|^2] \\
 &= \frac{1}{N} \mathcal{E} \left[\sum_{m=0}^{N-1} x(m) e^{-j2\pi f m} \sum_{n=0}^{N-1} x(n) e^{j2\pi f n} \right] \\
 &= \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N} \right) r_{xx}(m) e^{-j2\pi f m}
 \end{aligned} \tag{9.26}$$

As the number of signal samples N increases, we have

$$\lim_{N \rightarrow \infty} \mathcal{E}[\hat{P}_{XX}(f)] = \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi f m} = P_{XX}(f) \tag{9.27}$$

For a Gaussian random sequence, the variance of the periodogram can be obtained as

$$\text{Var}[\hat{P}_{XX}(f)] = P_{XX}^2(f) \left[1 + \left(\frac{\sin 2\pi f N}{N \sin 2\pi f} \right)^2 \right] \tag{9.28}$$

As the length of a signal record N increases, the expectation of the periodogram converges to the power spectrum $P_{XX}(f)$ and the variance of $\hat{P}_{XX}(f)$ converges to $P_{XX}^2(f)$. Hence the periodogram is an unbiased but not a consistent estimate.

The periodograms can be calculated from a DFT of the signal $x(m)$, or from a DFT of the autocorrelation estimates, $\hat{r}_{xx}(m)$. In addition, the signal from which the periodogram, or the autocorrelation samples, is obtained can be segmented into overlapping blocks to result in a larger number of periodograms, which can then be averaged. These methods and their effects on the variance of periodograms are considered in the following.

9.4.2 AVERAGING PERIODOGRAMS (BARTLETT METHOD)

In this method, several periodograms, from different segments of a signal, are averaged in order to reduce the variance of the periodogram. The Bartlett periodogram is obtained as the average of K periodograms as

$$\hat{P}_{XX}^B(f) = \frac{1}{K} \sum_{i=1}^K \hat{P}_{XX}^{(i)}(f) \tag{9.29}$$

where $\hat{P}_{XX}^{(i)}(f)$ is the periodogram of the i th segment of the signal. The expectation of the Bartlett periodogram $\hat{P}_{XX}^B(f)$ is given by

$$\begin{aligned} \mathcal{E}[\hat{P}_{XX}^B(f)] &= \mathcal{E}[\hat{P}_{XX}^{(i)}(f)] \\ &= \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N}\right) r_{xx}(m) e^{-j2\pi f m} \\ &= \frac{1}{N} \int_{-1/2}^{1/2} P_{XX}(\nu) \left[\frac{\sin \pi(f - \nu)N}{\sin \pi(f - \nu)} \right]^2 d\nu \end{aligned} \quad (9.30)$$

where $(\sin \pi f N / \sin \pi f)^2 / N$ is the frequency response of the triangular window $1 - |m|/N$. From Equation (9.30), the Bartlett periodogram is asymptotically unbiased. The variance of $\hat{P}_{XX}^B(f)$ is $1/K$ of the variance of the periodogram, and is given by

$$\text{Var} [\hat{P}_{XX}^B(f)] = \frac{1}{K} P_{XX}^2(f) \left[1 + \left(\frac{\sin 2\pi f N}{N \sin 2\pi f} \right)^2 \right] \quad (9.31)$$

9.4.3 WELCH METHOD: AVERAGING PERIODOGRAMS FROM OVERLAPPED AND WINDOWED SEGMENTS

In this method, a signal $x(m)$, of length M samples, is divided into K overlapping segments of length N , and each segment is windowed prior to computing the periodogram. The i th segment is defined as

$$x_i(m) = x(m + iD), \quad m = 0, \dots, N-1, \quad i = 0, \dots, K-1 \quad (9.32)$$

where D is the overlap. For half-overlap $D = N/2$, while $D = N$ corresponds to no overlap. For the i th windowed segment, the periodogram is given by

$$\hat{P}_{XX}^{(i)}(f) = \frac{1}{NU} \left| \sum_{m=0}^{N-1} w(m) x_i(m) e^{-j2\pi f m} \right|^2 \quad (9.33)$$

where $w(m)$ is the window function and U is the power in the window function, given by

$$U = \frac{1}{N} \sum_{m=0}^{N-1} w^2(m) \quad (9.34)$$

The spectrum of a finite-length signal typically exhibits side-lobes due to discontinuities at the endpoints. The window function, $w(m)$, alleviates the discontinuities and reduces the spread of the spectral energy into the side-lobes of the spectrum. The Welch power spectrum is the average of K periodograms obtained from overlapped and windowed segments of a signal:

$$\hat{P}_{XX}^W(f) = \frac{1}{K} \sum_{i=0}^{K-1} \hat{P}_{XX}^{(i)}(f) \quad (9.35)$$

Using Equations (9.33) and (9.35), the expectation of $\hat{P}_{XX}^W(f)$ can be obtained as

$$\begin{aligned}
 \mathcal{E}[P_{XX}^W(f)] &= \mathcal{E}[\hat{P}_{XX}^{(i)}(f)] \\
 &= \frac{1}{NU} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} w(n)w(m)\mathcal{E}[x_i(m)x_i(n)]e^{-j2\pi f(n-m)} \\
 &= \frac{1}{NU} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} w(n)w(m)r_{xx}(n-m)e^{-j2\pi f(n-m)} \\
 &= \int_{-1/2}^{1/2} P_{XX}(\nu)W(\nu-f) d\nu
 \end{aligned} \tag{9.36}$$

where

$$W(f) = \frac{1}{NU} \left| \sum_{m=0}^{N-1} w(m)e^{-j2\pi f m} \right|^2 \tag{9.37}$$

and the variance of the Welch estimate is given by

$$\text{Var}[\hat{P}_{XX}^W(f)] = \frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \mathcal{E}[\hat{P}_{XX}^{(i)}(f)\hat{P}_{XX}^{(j)}(f)] - \left\{ \mathcal{E}[\hat{P}_{XX}^W(f)] \right\}^2 \tag{9.38}$$

Welch has shown that, for the case when there is no overlap, $D = N$,

$$\text{Var}[P_{XX}^W(f)] = \frac{\text{Var}[P_{XX}^{(i)}(f)]}{K_1} \approx \frac{P_{XX}^2(f)}{K_1} \tag{9.39}$$

and for half-overlap, $D = N/2$,

$$\text{Var}[\hat{P}_{XX}^W(f)] = \frac{9}{8K_2} P_{XX}^2(f) \tag{9.40}$$

9.4.4 BLACKMAN–TUKEY METHOD

In this method, an estimate of a signal power spectrum is obtained from the Fourier transform of the windowed estimate of the autocorrelation function as

$$\hat{P}_{XX}^{\text{BT}}(f) = \sum_{m=-(N-1)}^{N-1} w(m)\hat{r}_{xx}(m)e^{-j2\pi f m} \tag{9.41}$$

For a signal of N samples, the number of samples available for estimation of the autocorrelation value at the lag m , $\hat{r}_{xx}(m)$, decreases as m approaches N . Therefore, for large m , the variance of the autocorrelation estimate increases, and the estimate becomes less

reliable. The window $w(m)$ has the effect of down-weighting the high variance coefficients at and around the end-points. The mean of the Blackman–Tukey power spectrum estimate is

$$\mathcal{E}[\hat{P}_{XX}^{\text{BT}}(f)] = \sum_{m=-(N-1)}^{N-1} \mathcal{E}[\hat{r}_{xx}(m)]w(m)e^{-j2\pi fm} \quad (9.42)$$

Now $\mathcal{E}[\hat{r}_{xx}(m)] = r_{xx}(m)w_B(m)$, where $w_B(m)$ is the Bartlett, or triangular, window. Equation (9.42) may be written as

$$\mathcal{E}[\hat{P}_{XX}^{\text{BT}}(f)] = \sum_{m=-(N-1)}^{N-1} r_{xx}(m)w_c(m)e^{-j2\pi fm} \quad (9.43)$$

where $w_c(m) = w_B(m)w(m)$. The right-hand side of Equation (9.43) can be written in terms of the Fourier transform of the autocorrelation and the window functions as

$$\mathcal{E}[\hat{P}_{XX}^{\text{BT}}(f)] = \int_{-1/2}^{1/2} P_{XX}(\nu)W_c(f-\nu)d\nu \quad (9.44)$$

where $W_c(f)$ is the Fourier transform of $w_c(m)$. The variance of the Blackman–Tukey estimate is given by

$$\text{Var}[\hat{P}_{XX}^{\text{BT}}(f)] \approx \frac{U}{N} P_{XX}^2(f) \quad (9.45)$$

where U is the energy of the window $w_c(m)$.

9.4.5 POWER SPECTRUM ESTIMATION FROM AUTOCORRELATION OF OVERLAPPED SEGMENTS

In the Blackman–Tukey method, in calculating a correlation sequence of length N from a signal record of length N , progressively fewer samples are admitted in estimation of $\hat{r}_{xx}(m)$ as the lag, m , approaches the signal length, N . Hence, the variance of $\hat{r}_{xx}(m)$, increases with the lag, m . This problem can be solved by using a signal of length $2N$ samples for calculation of N correlation values. In a generalization of this method, the signal record $x(m)$, of length M samples, is divided into a number K of overlapping segments of length $2N$. The i th segment is defined as

$$x_i(m) = x(m+iD), \quad m = 0, 1, \dots, 2N-1 \quad (9.46)$$

$$i = 0, 1, \dots, K-1$$

where D is the overlap. For each segment of length $2N$, the correlation function in the range of $0 \leq m \leq N$ is given by

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1} x_i(k)x_i(k+m), \quad m = 0, 1, \dots, N-1 \quad (9.47)$$

In Equation (9.47), the estimate of each correlation value is obtained as the averaged sum of N products.

9.5 MODEL-BASED POWER SPECTRUM ESTIMATION

In nonparametric power spectrum estimation, the autocorrelation function is assumed to be zero for lags $|m| \geq N$, beyond which no estimates are available. In parametric or model-based methods, a model of the signal process is used to extrapolate the autocorrelation function beyond the range $|m| \leq N$ for which data is available. Model-based spectral estimators have a better resolution than the periodograms, mainly because they do not assume that the correlation sequence is zero-valued for the range of lags for which no measurements are available.

In linear model-based spectral estimation, it is assumed that the signal $x(m)$ can be modelled as the output of a linear time-invariant system excited with a random, flat-spectrum, excitation. The assumption that the input has a flat spectrum implies that the power spectrum of the model output is *shaped* entirely by the frequency response of the model. The input-output relation of a generalized discrete linear time-invariant model is given by

$$x(m) = \sum_{k=1}^P a_k x(m-k) + \sum_{k=0}^Q b_k e(m-k) \quad (9.48)$$

where $x(m)$ is the model output, $e(m)$ is the input, and the a_k and b_k are the parameters of the model. Equation (9.48) is known as an autoregressive-moving-average (ARMA) model. The system function $H(z)$ of the discrete linear time-invariant model of Equation (9.48) is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^Q b_k z^{-k}}{1 - \sum_{k=1}^P a_k z^{-k}} \quad (9.49)$$

where $1/A(z)$ and $B(z)$ are the autoregressive and moving-average parts of $H(z)$, respectively. The power spectrum of the signal $x(m)$ is given as the product of the power spectrum of the input signal and the squared magnitude frequency response of the model:

$$P_{XX}(f) = P_{EE}(f) |H(f)|^2 \quad (9.50)$$

where $H(f)$ is the frequency response of the model and $P_{EE}(f)$ is the input power spectrum. Assuming that the input is a white noise process with unit variance, i.e. $P_{EE}(f) = 1$, Equation (9.50) becomes

$$P_{XX}(f) = |H(f)|^2 \quad (9.51)$$

Thus the power spectrum of the model output is the squared magnitude of the frequency response of the model. An important aspect of model-based spectral estimation is the choice of the model. The model may be an autoregressive (all-pole), a moving-average (all-zero) or an ARMA (pole-zero) model.

9.5.1 MAXIMUM-ENTROPY SPECTRAL ESTIMATION

The power spectrum of a stationary signal is defined as the Fourier transform of the autocorrelation sequence:

$$P_{XX}(f) = \sum_{n=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \quad (9.52)$$

Equation (9.52) requires the autocorrelation $r_{xx}(m)$ for the lag m in the range $\pm\infty$. In practice, an estimate of the autocorrelation $r_{xx}(m)$ is available only for the values of m in a finite range of, say, $\pm P$. In general, there are an infinite number of different correlation sequences that have the same values in the range $|m| \leq P$ as the measured values. The particular estimate used in the nonparametric methods assumes the correlation values are zero for the lags beyond $\pm P$, for which no estimates are available. This arbitrary assumption results in spectral leakage and loss of frequency resolution. *The maximum-entropy estimate is based on the principle that the estimate of the autocorrelation sequence must correspond to the most random signal whose correlation values in the range $|m| \leq P$ coincide with the measured values.*

The maximum-entropy principle is appealing because it assumes no more structure in the correlation sequence than that indicated by the measured data. The randomness or entropy of a signal is defined as

$$H[P_{XX}(f)] = \int_{-1/2}^{1/2} \ln P_{XX}(f) df \quad (9.53)$$

To obtain the maximum-entropy correlation estimate, we differentiate Equation (9.53) with respect to the unknown values of the correlation coefficients, and set the derivative to zero:

$$\frac{\partial H[P_{XX}(f)]}{\partial r_{xx}(m)} = \int_{-1/2}^{1/2} \frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)} df = 0 \quad \text{for } |m| > P \quad (9.54)$$

Now, from Equation (9.18), the derivative of the power spectrum with respect to the autocorrelation values is given by

$$\frac{\partial P_{XX}(f)}{\partial r_{xx}(m)} = e^{-j2\pi fm} \quad (9.55)$$

From Equation (9.55), for the derivative of the logarithm of the power spectrum, we have

$$\frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)} = P_{XX}^{-1}(f) e^{-j2\pi fm} \quad (9.56)$$

Substitution of Equation (9.56) in Equation (9.53) gives

$$\int_{-1/2}^{1/2} P_{XX}^{-1}(f) e^{-j2\pi fm} df = 0 \quad \text{for } |m| > P \quad (9.57)$$

Assuming that $P_{XX}^{-1}(f)$ is integrable, it may be associated with an autocorrelation sequence $c(m)$ as

$$P_{XX}^{-1}(f) = \sum_{m=-\infty}^{\infty} c(m) e^{-j2\pi fm} \quad (9.58)$$

where

$$c(m) = \int_{-1/2}^{1/2} P_{XX}^{-1}(f) e^{j2\pi fm} df \quad (9.59)$$

From Equations (9.57) and (9.59), we have $c(m) = 0$ for $|m| > P$. Hence, from Equation (9.58), the inverse of the maximum-entropy power spectrum may be obtained from the Fourier transform of a finite-length autocorrelation sequence as

$$P_{XX}^{-1}(f) = \sum_{m=-P}^P c(m) e^{-j2\pi fm} \quad (9.60)$$

and the maximum-entropy power spectrum is given by

$$\hat{P}_{XX}^{\text{ME}}(f) = \frac{1}{\sum_{m=-P}^P c(m) e^{-j2\pi fm}} \quad (9.61)$$

Since the denominator polynomial in Equation (9.61) is symmetric, it follows that, for every zero of this polynomial situated at a radius r , there is a zero at radius $1/r$. Hence this symmetric polynomial can be factorised and expressed as

$$\sum_{m=-P}^P c(m) z^{-m} = \frac{1}{\sigma^2} A(z) A(z^{-1}) \quad (9.62)$$

where $1/\sigma^2$ is a gain term, and $A(z)$ is a polynomial of order P defined as

$$A(z) = 1 + a_1 z^{-1} + \dots + a_p z^{-P} \quad (9.63)$$

From Equations (9.61) and (9.62), the maximum-entropy power spectrum may be expressed as

$$\hat{P}_{XX}^{\text{ME}}(f) = \frac{\sigma^2}{A(z) A(z^{-1})} \quad (9.64)$$

Equation (9.64) shows that the maximum-entropy power spectrum estimate is the power spectrum of an autoregressive model. Equation (9.64) was obtained by maximising the entropy of the power spectrum with respect to the unknown autocorrelation values. The known values of the autocorrelation function can be used to obtain the coefficients of the AR model of Equation (9.64), as discussed in the next section.

9.5.2 AUTOREGRESSIVE POWER SPECTRUM ESTIMATION

In the preceding section, it was shown that the maximum-entropy spectrum is equivalent to the spectrum of an autoregressive model of the signal. An autoregressive, or linear prediction model, described in detail in Chapter 8, is defined as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \quad (9.65)$$

where $e(m)$ is a random signal of variance σ_e^2 . The power spectrum of an autoregressive process is given by

$$P_{XX}^{\text{AR}}(f) = \frac{\sigma_e^2}{\left| 1 - \sum_{k=1}^P a_k e^{-j2\pi f k} \right|^2} \quad (9.66)$$

An AR model extrapolates the correlation sequence beyond the range for which estimates are available. The relation between the autocorrelation values and the AR model parameters is obtained by multiplying both sides of Equation (9.65) by $x(m-j)$ and taking the expectation:

$$\mathcal{E}[x(m)x(m-j)] = \sum_{k=1}^P a_k \mathcal{E}[x(m-k)x(m-j)] + \mathcal{E}[e(m)x(m-j)] \quad (9.67)$$

Now for the optimal model coefficients the random input $e(m)$ is orthogonal to the past samples, and Equation (9.67) becomes

$$r_{xx}(j) = \sum_{k=1}^P a_k r_{xx}(j-k), \quad j = 1, 2, \dots \quad (9.68)$$

Given $P+1$ correlation values, Equation (9.68) can be solved to obtain the AR coefficients a_k . Equation (9.68) can also be used to extrapolate the correlation sequence. The methods of solving the AR model coefficients are discussed in Chapter 8.

9.5.3 MOVING-AVERAGE POWER SPECTRUM ESTIMATION

A moving-average model is also known as an all-zero or a finite impulse response filter. A signal $x(m)$, modelled as a moving-average process, is described as

$$x(m) = \sum_{k=0}^Q b_k e(m-k) \quad (9.69)$$

where $e(m)$ is a zero-mean random input and Q is the model order. The cross-correlation of the input and output of a moving average process is given by

$$\begin{aligned} r_{xe}(m) &= \mathcal{E}[x(j)e(j-m)] \\ &= \mathcal{E}\left[\sum_{k=0}^Q b_k e(j-k)e(j-m) \right] = \sigma_e^2 b_m \end{aligned} \quad (9.70)$$

and the autocorrelation function of a moving average process is

$$r_{xx}(m) = \begin{cases} \sigma_c^2 \sum_{k=0}^{Q-|m|} b_k b_{k+m}, & |m| \leq Q \\ 0, & |m| > Q \end{cases} \quad (9.71)$$

From Equation (9.71), the power spectrum obtained from the Fourier transform of the autocorrelation sequence is the same as the power spectrum of a moving average model of the signal. Hence the power spectrum of a moving-average process may be obtained directly from the Fourier transform of the autocorrelation function as

$$P_{XX}^{\text{MA}} = \sum_{m=-Q}^Q r_{xx}(m) e^{-j2\pi f m} \quad (9.72)$$

Note that the moving-average spectral estimation is identical to the Blackman–Tukey method of estimating periodograms from the autocorrelation sequence.

9.5.4 AUTOREGRESSIVE MOVING-AVERAGE POWER SPECTRUM ESTIMATION

The ARMA, or pole–zero, model is described by Equation (9.48). The relationship between the ARMA parameters and the autocorrelation sequence can be obtained by multiplying both sides of Equation (9.48) by $x(m-j)$ and taking the expectation:

$$r_{xx}(j) = - \sum_{k=1}^P a_k r_{xx}(j-k) + \sum_{k=0}^Q b_k r_{xe}(j-k) \quad (9.73)$$

The moving-average part of Equation (9.73) influences the autocorrelation values only up to the lag of Q . Hence, for the autoregressive part of Equation (9.73), we have

$$r_{xx}(m) = - \sum_{k=1}^P a_k r_{xx}(m-k) \text{ for } m > Q \quad (9.74)$$

Hence Equation (9.74) can be used to obtain the coefficients a_k , which may then be substituted in Equation (9.73) for solving the coefficients b_k . Once the coefficients of an ARMA model are identified, the spectral estimate is given by

$$P_{XX}^{\text{ARMA}}(f) = \sigma_c^2 \frac{\left| \sum_{k=0}^Q b_k e^{-j2\pi f k} \right|^2}{\left| 1 + \sum_{k=1}^P a_k e^{-j2\pi f k} \right|^2} \quad (9.75)$$

where σ_c^2 is the variance of the input of the ARMA model. In general, the poles model the resonances of the signal spectrum, whereas the zeros model the antiresonances of the spectrum.

9.6 HIGH-RESOLUTION SPECTRAL ESTIMATION BASED ON SUBSPACE EIGENANALYSIS

The eigen-based methods considered in this section are primarily used for estimation of the parameters of sinusoidal signals observed in an additive white noise. Eigenanalysis is used for partitioning the eigenvectors and the eigenvalues of the autocorrelation matrix of a noisy signal into two subspaces:

- (1) the signal subspace composed of the *principle* eigenvectors associated with the largest eigenvalues;
- (2) the noise subspace represented by the smallest eigenvalues.

The decomposition of a noisy signal into a signal subspace and a noise subspace forms the basis of the eigenanalysis methods considered in this section.

9.6.1 PISARENKO HARMONIC DECOMPOSITION

A real-valued sine wave can be modelled by a second-order autoregressive model, with its poles on the unit circle at the angular frequency of the sinusoid, as shown in Figure 9.5. The AR model for a sinusoid of frequency F_i at a sampling rate of F_s is given by

$$x(m) = 2 \cos(2\pi F_i / F_s)x(m - 1) - x(m - 2) + A\delta(m - t_0) \tag{9.76}$$

where $A\delta(m - t_0)$ is the initial impulse for a sine wave of amplitude A . In general, a signal composed of P real sinusoids can be modelled by an AR model of order $2P$ as

$$x(m) = \sum_{k=1}^{2P} a_k x(m - k) + A\delta(m - t_0) \tag{9.77}$$

The transfer function of the AR model is given by

$$H(z) = \frac{A}{1 - \sum_{k=1}^{2P} a_k z^{-k}} = \frac{A}{\prod_{k=1}^P (1 - e^{-j2\pi F_k} z^{-1})(1 - e^{+j2\pi F_k} z^{-1})} \tag{9.78}$$

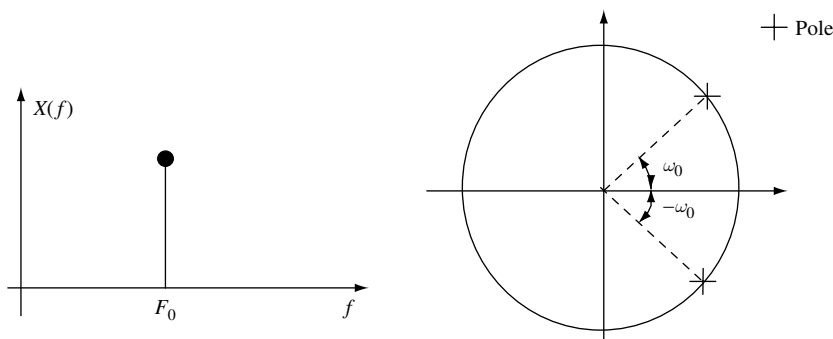


Figure 9.5 A second-order all-pole model of a sinusoidal signal.

where the angular positions of the poles on the unit circle, $e^{\pm j2\pi F_k}$, correspond to the angular frequencies of the sinusoids. For P real sinusoids observed in an additive white noise, we can write

$$\begin{aligned} y(m) &= x(m) + n(m) \\ &= \sum_{k=1}^{2P} a_k x(m-k) + n(m) \end{aligned} \quad (9.79)$$

Substituting $[y(m-k) - n(m-k)]$ for $x(m-k)$ in Equation (9.79) yields

$$y(m) - \sum_{k=1}^{2P} a_k y(m-k) = n(m) - \sum_{k=1}^{2P} a_k n(m-k) \quad (9.80)$$

From Equation (9.80), the noisy sinusoidal signal $y(m)$ can be modelled by an ARMA process in which the AR and the MA sections are identical, and the input is the noise process. Equation (9.80) can also be expressed in a vector notation as

$$\mathbf{y}^T \mathbf{a} = \mathbf{n}^T \mathbf{a} \quad (9.81)$$

where $\mathbf{y}^T = [y(m), \dots, y(m-2P)]$, $\mathbf{a}^T = [1, a_1, \dots, a_{2P}]$ and $\mathbf{n}^T = [n(m), \dots, n(m-2P)]$. To obtain the parameter vector \mathbf{a} , we multiply both sides of Equation (9.81) by the vector \mathbf{y} and take the expectation:

$$\mathcal{E}[\mathbf{y}\mathbf{y}^T] \mathbf{a} = \mathcal{E}[\mathbf{y}\mathbf{n}^T] \mathbf{a} \quad (9.82)$$

or

$$\mathbf{R}_{yy} \mathbf{a} = \mathbf{R}_{yn} \mathbf{a} \quad (9.83)$$

where $\mathcal{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{R}_{yy}$, and $\mathcal{E}[\mathbf{y}\mathbf{n}^T] = \mathbf{R}_{yn}$ can be written as

$$\begin{aligned} \mathbf{R}_{yn} &= \mathcal{E}[(\mathbf{x} + \mathbf{n})\mathbf{n}^T] \\ &= \mathcal{E}[\mathbf{n}\mathbf{n}^T] = \mathbf{R}_{nn} = \sigma_n^2 \mathbf{I} \end{aligned} \quad (9.84)$$

where σ_n^2 is the noise variance. Using Equation (9.84), Equation (9.83) becomes

$$\mathbf{R}_{yy} \mathbf{a} = \sigma_n^2 \mathbf{a} \quad (9.85)$$

Equation (9.85) is in the form of an eigenequation. If the dimension of the matrix \mathbf{R}_{yy} is greater than $2P \times 2P$, then the largest $2P$ eigenvalues are associated with the eigenvectors of the noisy sinusoids and the minimum eigenvalue corresponds to the noise variance σ_n^2 . The parameter vector \mathbf{a} is obtained as the eigenvector of \mathbf{R}_{yy} , with its first element unity and associated with the minimum eigenvalue. From the AR parameter vector \mathbf{a} , we can obtain the frequencies of the sinusoids by first calculating the roots of the polynomial

$$1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{2P} z^{-2P+2} + a_1 z^{-2P+1} - z^{-2P} = 0 \quad (9.86)$$

Note that for sinusoids, the AR parameters form a symmetric polynomial, that is $a_k = a_{2P-k}$. The frequencies F_k of the sinusoids can be obtained from the roots, z_k , of Equation (9.86) using the relation

$$z_k = e^{j2\pi F_k} \quad (9.87)$$

The powers of the sinusoids are calculated as follows. For P sinusoids observed in additive white noise, the autocorrelation function is given by

$$r_{yy}(k) = \sum_{i=1}^P P_i \cos 2k\pi F_i + \sigma_n^2 \delta(k) \quad (9.88)$$

where $P_i = A_i^2/2$ is the power of the sinusoid $A_i \sin(2\pi F_i)$, and white noise affects only the correlation at lag zero $r_{yy}(0)$. Hence Equation (9.88) for the correlation lags $k = 1, \dots, P$ can be written as

$$\begin{pmatrix} \cos 2\pi F_1 & \cos 2\pi F_2 & \dots & \cos 2\pi F_P \\ \cos 4\pi F_1 & \cos 4\pi F_2 & \dots & \cos 4\pi F_P \\ \vdots & \vdots & \ddots & \vdots \\ \cos 2P\pi F_1 & \cos 2P\pi F_2 & \dots & \cos 2P\pi F_P \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_P \end{pmatrix} = \begin{pmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \vdots \\ r_{yy}(P) \end{pmatrix} \quad (9.89)$$

Given an estimate of the frequencies F_i from Equations (9.86) and (9.87), and an estimate of the autocorrelation function $\hat{r}_{yy}(k)$, Equation (9.89) can be solved to obtain the powers of the sinusoids P_i . The noise variance can then be obtained from Equation (9.88) as

$$\sigma_n^2 = r_{yy}(0) - \sum_{i=1}^P P_i \quad (9.90)$$

9.6.2 MULTIPLE SIGNAL CLASSIFICATION SPECTRAL ESTIMATION

The multiple signal classification (MUSIC) algorithm is an eigen-based subspace decomposition method for estimation of the frequencies of complex sinusoids observed in additive white noise. Consider a signal $y(m)$ modelled as

$$y(m) = \sum_{k=1}^P A_k e^{-j(2\pi F_k m + \phi_k)} + n(m) \quad (9.91)$$

An N -sample vector $\mathbf{y} = [y(m), \dots, y(m+N-1)]$ of the noisy signal can be written as

$$\begin{aligned} \mathbf{y} &= \mathbf{x} + \mathbf{n} \\ &= \mathbf{S}\mathbf{a} + \mathbf{n} \end{aligned} \quad (9.92)$$

where the signal vector $\mathbf{x} = \mathbf{S}\mathbf{a}$ is defined as

$$\begin{pmatrix} x(m) \\ x(m+1) \\ \vdots \\ x(m+N-1) \end{pmatrix} = \begin{pmatrix} e^{j2\pi F_1 m} & e^{j2\pi F_2 m} & \dots & e^{j2\pi F_P m} \\ e^{j2\pi F_1 (m+1)} & e^{j2\pi F_2 (m+1)} & \dots & e^{j2\pi F_P (m+1)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j2\pi F_1 (m+N-1)} & e^{j2\pi F_2 (m+N-1)} & \dots & e^{j2\pi F_P (m+N-1)} \end{pmatrix} \begin{pmatrix} A_1 e^{j2\pi \phi_1} \\ A_2 e^{j2\pi \phi_2} \\ \vdots \\ A_P e^{j2\pi \phi_P} \end{pmatrix} \quad (9.93)$$

The matrix \mathbf{S} and the vector \mathbf{a} are defined on the right-hand side of Equation (9.93). The autocorrelation matrix of the noisy signal \mathbf{y} can be written as the sum of the autocorrelation matrices of the signal \mathbf{x} and the noise as

$$\begin{aligned}\mathbf{R}_{yy} &= \mathbf{R}_{xx} + \mathbf{R}_{nn} \\ &= \mathbf{S}\mathbf{P}\mathbf{S}^H + \sigma_n^2\mathbf{I}\end{aligned}\quad (9.94)$$

where $\mathbf{R}_{xx} = \mathbf{S}\mathbf{P}\mathbf{S}^H$ and $\mathbf{R}_{nn} = \sigma_n^2\mathbf{I}$ are the autocorrelation matrices of the signal and noise processes, the exponent H denotes the Hermitian transpose, and the diagonal matrix \mathbf{P} defines the power of the sinusoids as

$$\mathbf{P} = \mathbf{a}\mathbf{a}^H = \text{diag}[P_1, P_2, \dots, P_P] \quad (9.95)$$

where $P_i = A_i^2$ is the power of the complex sinusoid $e^{-j2\pi F_i}$. The correlation matrix of the signal can also be expressed in the form

$$\mathbf{R}_{xx} = \sum_{k=1}^P P_k \mathbf{s}_k \mathbf{s}_k^H \quad (9.96)$$

where $\mathbf{s}_k^H = [1, e^{j2\pi F_k}, \dots, e^{j2\pi(N-1)F_k}]$. Now consider an eigendecomposition of the $N \times N$ correlation matrix \mathbf{R}_{xx}

$$\begin{aligned}\mathbf{R}_{xx} &= \sum_{k=1}^N \lambda_k \mathbf{v}_k \mathbf{v}_k^H \\ &= \sum_{k=1}^P \lambda_k \mathbf{v}_k \mathbf{v}_k^H\end{aligned}\quad (9.97)$$

where λ_k and \mathbf{v}_k are the eigenvalues and eigenvectors of the matrix \mathbf{R}_{xx} , respectively. We have also used the fact that the autocorrelation matrix, \mathbf{R}_{xx} , of P complex sinusoids has only P nonzero eigenvalues, $\lambda_{P+1} = \lambda_{P+2}, \dots, \lambda_N = 0$. Since the sum of the cross-products of the eigenvectors forms an identity matrix, we can also express the diagonal autocorrelation matrix of the noise in terms of the eigenvectors of \mathbf{R}_{xx} as

$$\mathbf{R}_{nn} = \sigma_n^2 \mathbf{I} = \sigma_n^2 \sum_{k=1}^N \mathbf{v}_k \mathbf{v}_k^H \quad (9.98)$$

The correlation matrix of the noisy signal may be expressed in terms of its eigenvectors and the associated eigenvalues of the noisy signal as

$$\begin{aligned}\mathbf{R}_{yy} &= \sum_{k=1}^P \lambda_k \mathbf{v}_k \mathbf{v}_k^H + \sigma_n^2 \sum_{k=1}^N \mathbf{v}_k \mathbf{v}_k^H \\ &= \sum_{k=1}^P (\lambda_k + \sigma_n^2) \mathbf{v}_k \mathbf{v}_k^H + \sigma_n^2 \sum_{k=P+1}^N \mathbf{v}_k \mathbf{v}_k^H\end{aligned}\quad (9.99)$$

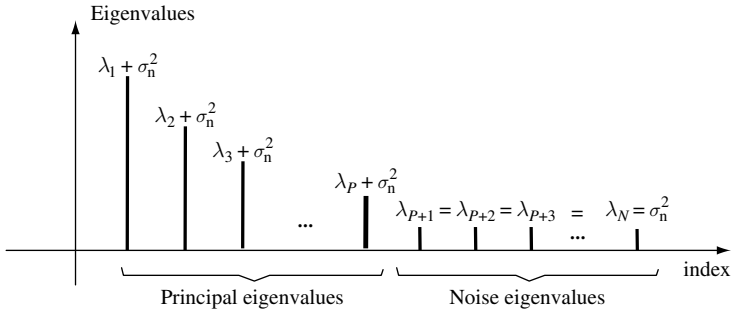


Figure 9.6 Decomposition of the eigenvalues of a noisy signal into the principal eigenvalues and noise eigenvalues.

From Equation (9.99), the eigenvectors and the eigenvalues of the correlation matrix of the noisy signal can be partitioned into two disjoint subsets (see Figure 9.6). The set of eigenvectors $\{\mathbf{v}_1, \dots, \mathbf{v}_P\}$ associated with the P largest eigenvalues spans the ‘signal subspace’ and they are called the ‘principal eigenvectors’. The signal vectors \mathbf{s}_i can be expressed as linear combinations of the principal eigenvectors. The second subset of eigenvectors $\{\mathbf{v}_{P+1}, \dots, \mathbf{v}_N\}$ spans the ‘noise subspace’ and they have σ_n^2 as their eigenvalues. Since the signal and noise eigenvectors are orthogonal, it follows that the signal subspace and the noise subspace are orthogonal. Hence the sinusoidal signal vectors, \mathbf{s}_i , which are in the signal subspace, are orthogonal to the noise subspace, and we have

$$\mathbf{s}_i^H(f)\mathbf{v}_k = \sum_{m=0}^{N-1} v_k(m)e^{-j2\pi F_i m} = 0 \quad i = 1, \dots, P \quad k = P+1, \dots, N \quad (9.100)$$

Equation (9.100) implies that the frequencies of the P sinusoids can be obtained by solving for the zeros of the following polynomial function of the frequency variable, f :

$$\sum_{k=P+1}^N \mathbf{s}^H(f)\mathbf{v}_k \quad (9.101)$$

In the MUSIC algorithm, the power spectrum estimate is defined as

$$P_{XX}(f) = \sum_{k=P+1}^N |\mathbf{s}^H(f)\mathbf{v}_k|^2 \quad (9.102)$$

where $\mathbf{s}(f) = [1, e^{j2\pi f}, \dots, e^{j2\pi(N-1)f}]$ is the complex sinusoidal vector, and $\{\mathbf{v}_{P+1}, \dots, \mathbf{v}_N\}$ are the eigenvectors in the noise subspace. From Equations (9.100) and (9.102) we have that

$$P_{XX}(f_i) = 0, \quad i = 1, \dots, P \quad (9.103)$$

Since $P_{XX}(f)$ has its zeros at the frequencies of the sinusoids, it follows that the reciprocal of $P_{XX}(f)$ has its poles at these frequencies. The MUSIC spectrum is defined as

$$P_{XX}^{\text{MUSIC}}(f) = \frac{1}{\sum_{k=P+1}^N |\mathbf{s}^H(f)\mathbf{v}_k|^2} = \frac{1}{\mathbf{s}^H(f)\mathbf{V}(f)\mathbf{V}^H(f)\mathbf{s}(f)} \quad (9.104)$$

where $\mathbf{V} = [\mathbf{v}_{p+1}, \dots, \mathbf{v}_N]$ is the matrix of eigenvectors of the noise subspace. $P_{\text{MUSIC}}(f)$ is sharply peaked at the frequencies of the sinusoidal components of the signal, and hence the frequencies of its peaks are taken as the MUSIC estimates.

9.6.3 ESTIMATION OF SIGNAL PARAMETERS VIA ROTATIONAL INVARIANCE TECHNIQUES

The estimation of signal parameters via rotational invariance techniques (ESPIRIT) algorithm is an eigendecomposition approach for estimating the frequencies of a number of complex sinusoids observed in additive white noise. Consider a signal, $y(m)$, composed of P complex-valued sinusoids and additive white noise:

$$y(m) = \sum_{k=1}^P A_k e^{-j(2\pi F_k m + \phi_k)} + n(m) \quad (9.105)$$

The ESPIRIT algorithm exploits the deterministic relation between sinusoidal component of the signal vector, $\mathbf{y}(m) = [y(m), \dots, y(m+N-1)]^T$, and that of the time-shifted vector, $\mathbf{y}(m+1) = [y(m+1), \dots, y(m+N)]^T$. The signal component of the noisy vector $\mathbf{y}(m)$ may be expressed as

$$\mathbf{x}(m) = \mathbf{S}\mathbf{a} \quad (9.106)$$

where \mathbf{S} is the complex sinusoidal matrix and \mathbf{a} is the vector containing the amplitude and phase of the sinusoids as in Equations (9.92) and (9.93). A complex sinusoid $e^{j2\pi F_i m}$ can be time-shifted by one sample through multiplication by a phase term, $e^{j2\pi F_i}$. Hence the time-shifted sinusoidal signal vector, $\mathbf{x}(m+1)$, may be obtained from $\mathbf{x}(m)$ by phase-shifting each complex sinusoidal component of $\mathbf{x}(m)$ as

$$\mathbf{x}(m+1) = \mathbf{S}\Phi\mathbf{a} \quad (9.107)$$

where Φ is a $P \times P$ phase matrix defined as

$$\Phi = \text{diag}[e^{j2\pi F_1}, e^{j2\pi F_2}, \dots, e^{j2\pi F_P}] \quad (9.108)$$

The diagonal elements of Φ are the relative phases between the adjacent samples of the sinusoids. The matrix Φ is a unitary matrix and is known as a ‘rotation matrix’ since it relates the time-shifted vectors $\mathbf{x}(m)$ and $\mathbf{x}(m+1)$. The autocorrelation matrix of the noisy signal vector $\mathbf{y}(m)$ can be written as

$$\mathbf{R}_{\mathbf{y}(m)\mathbf{y}(m)} = \mathbf{S}\mathbf{P}\mathbf{S}^H + \sigma_n^2 \mathbf{I} \quad (9.109)$$

where the matrix \mathbf{P} is diagonal, and its diagonal elements are the powers of the complex sinusoids, $\mathbf{P} = \text{diag}[A_1^2, \dots, A_P^2] = \mathbf{a}\mathbf{a}^H$. The cross-covariance matrix of the vectors $\mathbf{y}(m)$ and $\mathbf{y}(m+1)$ is

$$\mathbf{R}_{\mathbf{y}(m)\mathbf{y}(m+1)} = \mathbf{S}\mathbf{P}\Phi^H\mathbf{S}^H + \mathbf{R}_{\mathbf{n}(m)\mathbf{n}(m+1)} \quad (9.110)$$

where the autocovariance matrices $\mathbf{R}_{y(m)y(m+1)}$ and $\mathbf{R}_{n(m)n(m+1)}$ are defined as

$$\mathbf{R}_{y(m)y(m+1)} = \begin{pmatrix} r_{yy}(1) & r_{yy}(2) & r_{yy}(3) & \cdots & r_{yy}(N) \\ r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \cdots & r_{yy}(N-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \cdots & r_{yy}(N-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(N-2) & r_{yy}(N-3) & r_{yy}(N-4) & \cdots & r_{yy}(1) \end{pmatrix} \quad (9.111)$$

and

$$\mathbf{R}_{n(m)n(m+1)} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \sigma_n^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_n^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 & 0 \end{pmatrix} \quad (9.112)$$

The correlation matrix of the signal vector $\mathbf{x}(m)$ can be estimated as

$$\mathbf{R}_{\mathbf{x}(m)\mathbf{x}(m)} = \mathbf{R}_{y(m)y(m)} - \mathbf{R}_{n(m)n(m)} = \mathbf{SPS}^H \quad (9.113)$$

and the cross-correlation matrix of the signal vector $\mathbf{x}(m)$ with its time-shifted version $\mathbf{x}(m+1)$ is obtained as

$$\mathbf{R}_{\mathbf{x}(m)\mathbf{x}(m+1)} = \mathbf{R}_{y(m)y(m+1)} - \mathbf{R}_{n(m)n(m+1)} = \mathbf{SP}\Phi^H\mathbf{S}^H \quad (9.114)$$

Subtraction of a fraction $\lambda_i = e^{-j2\pi F_i}$ of Equation (9.112) from Equation (9.113) yields

$$\mathbf{R}_{\mathbf{x}(m)\mathbf{x}(m)} - \lambda_i \mathbf{R}_{\mathbf{x}(m)\mathbf{x}(m+1)} = \mathbf{SP}(\mathbf{I} - \lambda_i \Phi^H)\mathbf{S}^H \quad (9.115)$$

From Equations (9.108) and (9.115), the frequencies of the sinusoids can be estimated as the roots of Equation (9.115).

9.7 SUMMARY

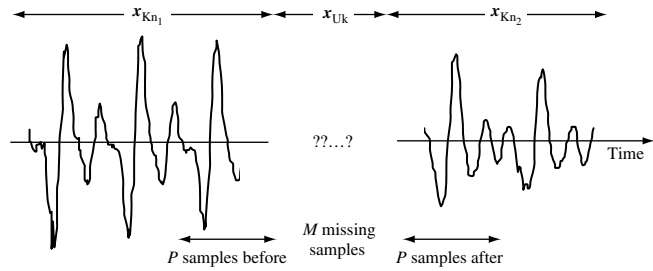
Power spectrum estimation is perhaps the most widely used method of signal analysis. The main objective of any transformation is to express a signal in a form that lends itself to more convenient analysis and manipulation. The power spectrum is related to the correlation function through the Fourier transform. The power spectrum reveals the repetitive and correlated patterns of a signal, which are important in detection, estimation, data forecasting and decision-making systems. We began this chapter with Section 9.1 on basic definitions of the Fourier series/transform, energy spectrum and power spectrum. In Section 9.2, we considered nonparametric DFT-based methods of spectral analysis. These methods do not offer the high resolution of parametric and eigen-based methods. However, they are attractive in that they are computationally less expensive than model-based methods and are relatively robust. In Section 9.3, we considered the maximum-entropy and the model-based spectral

estimation methods. These methods can extrapolate the correlation values beyond the range for which data is available, and hence can offer higher resolution and fewer side-lobes. In Section 9.4, we considered the eigen-based spectral estimation of noisy signals. These methods decompose the eigenvariables of the noisy signal into a signal subspace and a noise subspace. The orthogonality of the signal and noise subspaces is used to estimate the signal and noise parameters. In the next chapter, we use DFT-based spectral estimation for restoration of signals observed in noise.

BIBLIOGRAPHY

- Bartlett M.S. (1950) Periodogram analysis and continuous spectra. *Biometrika*, **37**: 1–16.
- Blackman R.B. and Tukey J.W. (1958) *The Measurement of Power Spectra from the Point of View of Communication Engineering*. Dover, New York.
- Bracewell R.N. (1965) *The Fourier Transform and its Applications*. McGraw–Hill, New York.
- Brault, J.W. and White, O.R. (1971) The analysis and restoration of astronomical data via the fast Fourier transform. *Astron. Astrophys.*, **13**: 169–189.
- Brigham, E. (1988) *The Fast Fourier Transform and its Applications*. Englewood Cliffs, Prentice-Hall, NJ.
- Burg J.P. (1975) Maximum entropy spectral analysis. Ph.D. thesis, Department of Geophysics, Stanford University, Stanford, CA.
- Cadzow J.A. (1979) ARMA spectral estimation: an efficient closed-form procedure. *Proceedings of RADC Spectrum Estimation Workshop*, pp. 81–97.
- Capon J. (1969) High resolution frequency-wavenumber spectrum analysis. *Proc. IEEE*, **57**: 1408–1419.
- Childers D.G. (ed.) (1978) *Modern Spectrum Analysis*. IEEE Press, New York.
- Cohen L. (1989) Time-frequency distributions – a review. *Proc. IEEE*, **77**: 941–981.
- Cooley, J.W. and Tukey, J.W. (1965) An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, **19**(90): 297–301.
- Fourier, J.B.J. (1878) *Théorie Analytique de la Chaleur*, Trans. Alexander Freeman. Reprinted by Dover, New York, 1955.
- Grattam-Guinness I. (1972) *Joseph Fourier (1768–1830): a Survey of His Life and Work*. MIT Press, Cambridge, MA.
- Haykin S. (1985) *Array Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.
- Jenkins G.M. and Watts D.G. (1968) *Spectral Analysis and its Applications*. Holden-Day, San Francisco, CA.
- Kay S.M. (1988) *Modern Spectral Estimation: Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ.
- Kay S.M. and Marple S.L. (1981) Spectrum analysis: a modern perspective. *Proc. IEEE*, **69**: 1380–1419.
- Lacoss R.T. (1971) Data adaptive spectral analysis methods. *Geophysics*, **36**: 661–675.
- Marple S.L. (1987) *Digital Spectral Analysis with Applications*. Prentice Hall, Englewood Cliffs, NJ.
- Parzen E. (1957) On consistent estimates of the spectrum of a stationary time series. *Am. Math. Statist.*, **28**: 329–349.
- Pisarenko V.F. (1973) The retrieval of harmonics from a covariance function. *Geophys J. R. Astron. Soc.*, **33**: 347–366.
- Roy R. H. (1987) ESPRIT – estimation of signal parameters via rotational invariance techniques. Ph.D. thesis, Stanford University, Stanford, CA.
- Schmidt R.O. (1981) A signal subspace approach to multiple emitter location and spectral estimation. Ph.D. thesis, Stanford University, Stanford, CA.
- Stanislav B.K. (ed.) (1986) *Modern Spectrum Analysis*. IEEE Press, New York.
- Strand O.N. (1977) Multichannel complex maximum entropy (autoregressive) spectral analysis. *IEEE Trans. Autom. Control*, **22**(4): 634–640.
- Van Den Bos A. (1971) Alternative interpretation of maximum entropy spectral analysis. *IEEE Trans. Inform. Technol.*, **IT-17**: 92–99.
- Welch P.D. (1967) The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short modified periodograms. *IEEE Trans. Audio Electroacoust.*, **AU-15**: 70–79.
- Wilkinson J.H. (1965) *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford.

10



Interpolation

10.1 Introduction	268
10.2 Polynomial Interpolation	274
10.3 Model-Based Interpolation	280
10.4 Summary	294
Bibliography	295

Interpolation is the estimation of the unknown, or the lost, samples of a signal using a weighted average of a number of known samples at the neighbourhood points. Interpolators are used in various forms in most communications signal processing and decision-making systems. Applications of interpolators include conversion of a discrete-time signal to a continuous-time signal, sampling rate conversion in multirate communications systems, low-bit-rate speech coding, up-sampling of a signal for improved graphical representation and restoration of a sequence of samples irrevocably distorted by transmission errors, packet loss, impulsive noise, dropouts, etc.

This chapter begins with a study of the basic concepts of ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples and the factors that affect the interpolation process. The classical approach to interpolation is to construct a polynomial that passes through the known samples. In Section 10.2, a general form of polynomial interpolation and its special forms, Lagrange, Newton, Hermite and cubic spline interpolators, are considered.

Optimal interpolators utilise predictive models of signal trajectory and statistical models of the distribution of the signal process. In Section 10.3, a number of model-based interpolation

methods are considered. These methods include maximum *a posteriori* interpolation and least square error interpolation based on an autoregressive model. Finally, we consider time–frequency interpolation, and interpolation through searching an adaptive signal codebook for the best-matching signal.

10.1 INTRODUCTION

The objective of interpolation is to obtain a high-fidelity reconstruction of the unknown or the missing samples of a signal. The emphasis in this chapter is on the interpolation of a *sequence* of lost samples. However, first in this section, the basic theory of ideal interpolation of a band-limited signal is introduced, and its applications in conversion of a discrete-time signal to a continuous-time signal and in conversion of the sampling rate of a digital signal are considered. Then a simple distortion model is used to gain insight into the effects of a sequence of lost samples and the methods of recovery of the lost samples. The factors that affect interpolation error are also considered in this section.

10.1.1 INTERPOLATION OF A SAMPLED SIGNAL

A common application of interpolation is the reconstruction of a continuous-time signal, $x(t)$, from a discrete-time signal, $x(m)$. The condition for the recovery of a continuous-time signal from its samples is given by the Nyquist sampling theorem. The Nyquist theorem states that a band-limited signal, with a highest frequency content of F_c (Hz), can be reconstructed from its samples *if* the sampling speed is greater than $2F_c$ samples per second. Consider a band-limited continuous-time signal $x(t)$, sampled at a rate of F_s samples per second. The discrete-time signal, $x(m)$, may be expressed as the following product:

$$x(m) = x(t)p(t) = \sum_{m=-\infty}^{\infty} x(t)\delta(t - mT_s) \quad (10.1)$$

where $p(t) = \sum \delta(t - mT_s)$ is the sampling function and $T_s = 1/F_s$ is the sampling interval. Taking the Fourier transform of Equation (10.1), it can be shown that the spectrum of the sampled signal is given by

$$X_s(f) = X(f) * P(f) = \sum_{k=-\infty}^{\infty} X(f + kf_s) \quad (10.2)$$

where $X(f)$ and $P(f)$ are the spectra of the signal $x(t)$ and the sampling function $p(t)$, respectively, and $*$ denotes the convolution operation.

Equation (10.2), illustrated in Figure 10.1, states that the spectrum of a sampled signal is composed of the original base-band spectrum $X(f)$ and the repetitions or images of $X(f)$ spaced uniformly at frequency intervals of $F_s = 1/T_s$. When the sampling frequency is above the Nyquist rate, the base-band spectrum, $X(f)$, is not overlapped by its images, $X(f \pm kF_s)$, and the original signal can be recovered by a low-pass filter, as shown in Figure 10.1. Hence the ideal interpolator of a band-limited discrete-time signal is an ideal low-pass filter with a

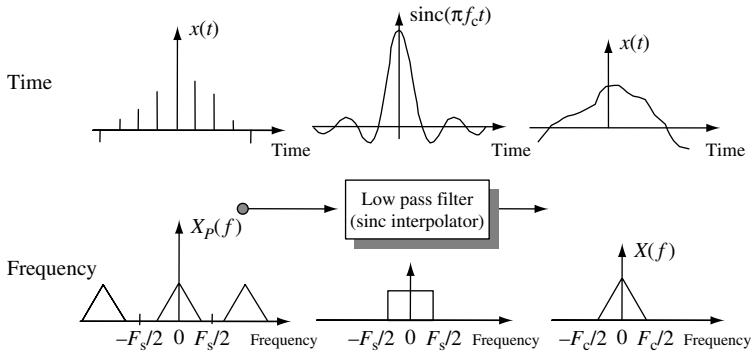


Figure 10.1 Reconstruction of a continuous-time signal from its discrete-time samples. In the frequency domain interpolation is equivalent to low-pass filtering.

sinc impulse response. The recovery of a continuous-time signal through sinc interpolation can be expressed as

$$x(t) = \sum_{m=-\infty}^{\infty} x(m)T_s f_c \text{sinc} [\pi f_c (t - mT_s)] \tag{10.3}$$

In practice, the sampling rate F_s should be sufficiently greater than $2F_c$, say $2.5F_c$, in order to accommodate the transition bandwidth of the interpolating low-pass filter.

10.1.2 DIGITAL INTERPOLATION BY A FACTOR OF I

Applications of digital interpolators include sampling rate conversion in multirate communication systems and up-sampling for improved graphical representation. To change a sampling rate by a factor of $V = I/D$ (where I and D are integers), the signal is first interpolated by a factor of I , and then the interpolated signal is decimated by a factor of D .

Consider a band-limited discrete-time signal, $x(m)$, with a base-band spectrum, $X(f)$, as shown in Figure 10.2. The sampling rate can be increased by a factor of I through interpolation of $I - 1$ samples between every two samples of $x(m)$. In the following it is shown that digital interpolation by a factor of I can be achieved through a two-stage process of (a) insertion of $I - 1$ zeros in between every two samples and (b) low-pass filtering of the

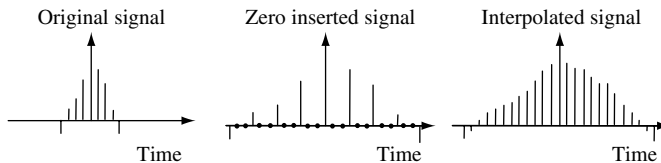


Figure 10.2 Illustration of up-sampling by a factor of 3 using a two-stage process of zero-insertion and digital low-pass filtering.

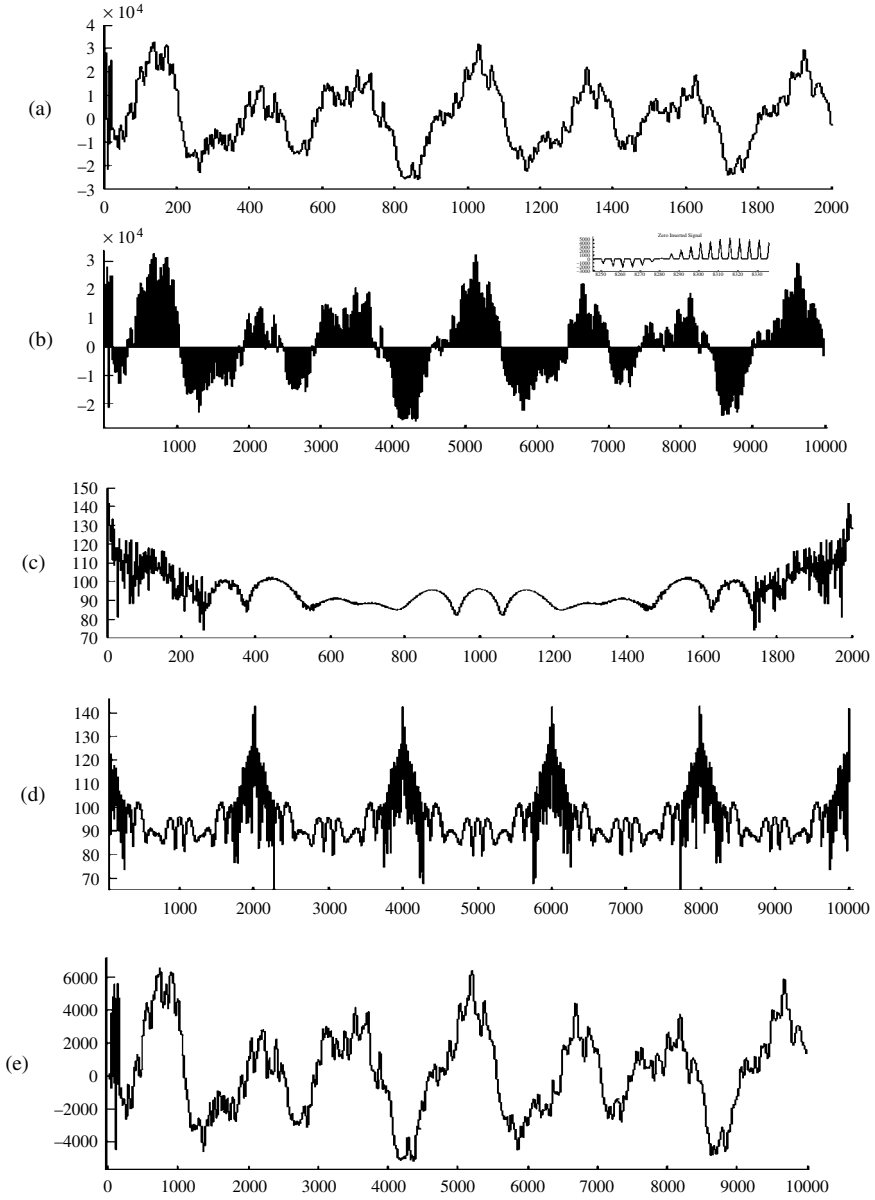


Figure 10.3 (a) Original signal; (b) zero-inserted signal; (c) spectrum of original signal; (d) spectrum of zero-inserted signal; (e) interpolated signal.

zero-inserted signal by a filter with a cutoff frequency of $F_s/2I$, where F_s is the sampling rate. Consider the zero-inserted signal $x_z(m)$ obtained by inserting $I - 1$ zeros between every two samples of $x(m)$ and expressed as

$$x_z(m) = \begin{cases} x\left(\frac{m}{I}\right), & m = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases} \quad (10.4)$$

The spectrum of the zero-inserted signal is related to the spectrum of the original discrete-time signal by

$$\begin{aligned} X_z(f) &= \sum_{m=-\infty}^{\infty} x_z(m) e^{-j2\pi f m} \\ &= \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi f m I} \\ &= X(I, f) \end{aligned} \quad (10.5)$$

Equation (10.5) states that the spectrum of the zero-inserted signal $X_z(f)$ is a frequency-scaled version of the spectrum of the original signal $X(f)$.

The process of interpolation of a signal through zero-insertion and low pass filtering is illustrated in Figure 10.3. A segment of speech and its zero-inserted version are shown in Figures 10.3(a) and 10.3(b) respectively. Figures 10.3(c) and (d) show the base-band spectra of the original and the zero-inserted speech respectively. Note that spectrum of the zero-inserted signal is composed of I (in this case $I = 5$) repetitions of the based band spectrum of the original signal. Interpolation of the zero-inserted signal is achieved by filtering out the repetitions of $X(f)$ in the base band of $X_z(f)$. The interpolated signal is shown in Figure 10.3(e). Note that during the playback, to maintain the real-time duration of the signal, the sampling rate of the interpolated signal $x_z(m)$ needs to be increased by a factor of I .

10.1.3 INTERPOLATION OF A SEQUENCE OF LOST SAMPLES

In this section, we introduce the problem of interpolation of a sequence of M missing samples of a signal given a number of samples on both sides of the gap, as illustrated in Figure 10.4. Perfect interpolation, with zero error, is only possible if the missing samples are redundant, in the sense that they carry no more information than that conveyed by the known neighbouring samples. This will be the case if the signal is a perfectly predictable signal such as a sine wave,

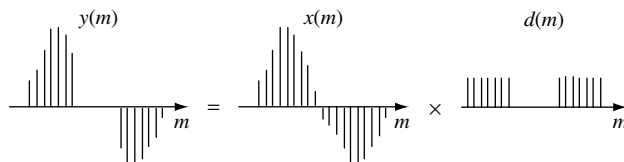


Figure 10.4 Illustration of a distortion model for a signal with a sequence of missing samples.

or in the case of a band-limited random signal if the sampling rate is greater than M times the Nyquist rate. However, in many practical cases, the signal is a realisation of a random process, and the sampling rate is only marginally above the Nyquist rate. In such cases, the lost samples cannot be perfectly recovered, and some interpolation error is inevitable.

A simple distortion model for a signal $y(m)$ with M missing samples, illustrated in Figure 10.4, is given by

$$\begin{aligned} y(m) &= x(m)d(m) \\ &= x(m)[1 - r(m)] \end{aligned} \quad (10.6)$$

where the distortion operator $d(m)$ is defined as

$$d(m) = 1 - r(m) \quad (10.7)$$

and $r(m)$ is a rectangular pulse of duration M samples starting at the sampling time k :

$$r(m) = \begin{cases} 1, & k \leq m \leq k + M - 1 \\ 0, & \text{otherwise} \end{cases} \quad (10.8)$$

In the frequency domain, Equation (10.6) becomes

$$\begin{aligned} Y(f) &= X(f)*D(f) \\ &= X(f)*[\delta(f) - R(f)] \\ &= X(f) - X(f)*R(f) \end{aligned} \quad (10.9)$$

where $D(f)$ is the spectrum of the distortion $d(m)$, $\delta(f)$ is the Kronecker delta function, and $R(f)$, the frequency spectrum of the rectangular pulse $r(m)$, is given by

$$R(f) = e^{-j2\pi f[k+(M-1)/2]} \frac{\sin(\pi fM)}{\sin(\pi f)} \quad (10.10)$$

In general, the distortion $d(m)$ is a noninvertible, many-to-one transformation, and perfect interpolation with zero error is not possible. However, as discussed in Section 10.3, the interpolation error can be minimised through optimal utilisation of the signal models and the information contained in the neighbouring samples.

Example 10.1: Interpolation of Missing Samples of a Sinusoidal Signal

Consider a cosine waveform of amplitude A and frequency F_0 with M missing samples, modelled as

$$\begin{aligned} y(m) &= x(m)d(m) \\ &= A(\cos 2\pi f_0 m)[1 - r(m)b] \end{aligned} \quad (10.11)$$

where $r(m)$ is the rectangular pulse defined in Equation (10.8). In the frequency domain, the distorted signal can be expressed as

$$\begin{aligned} Y(f) &= \frac{A}{2} [\delta(f - f_0) + \delta(f + f_0)]^* [\delta(f) - R(f)] \\ &= \frac{A}{2} [\delta(f - f_0) + \delta(f + f_0) - R(f - f_0) - R(f + f_0)] \end{aligned} \quad (10.12)$$

where $R(f)$ is the spectrum of the pulse $r(m)$, as in Equation (10.9).

From Equation (10.12), it is evident that, for a cosine signal of frequency F_0 , the distortion in the frequency domain due to the missing samples is manifested in the appearance of sinc functions centred at $\pm F_0$. The distortion can be removed by filtering the signal with a very narrow band-pass filter. Note that, for a cosine signal, perfect restoration is possible only because the signal has infinitely narrow bandwidth, or equivalently because the signal is completely predictable. In fact, for this example, the distortion can also be removed using a linear prediction model, which, for a cosine signal, can be regarded as a data-adaptive narrow-bandpass filter.

10.1.4 THE FACTORS THAT AFFECT INTERPOLATION ACCURACY

The interpolation accuracy is affected by a number of factors, the most important of which are as follows:

- (1) The predictability, or correlation structure, of the signal – as the correlation of successive samples increases, the predictability of a sample from the neighbouring samples increases. In general, interpolation improves with increasing correlation structure, or equivalently decreasing bandwidth, of a signal.
- (2) The sampling rate – as the sampling rate increases, adjacent samples become more correlated, the redundant information increases, and interpolation improves.
- (3) Nonstationary characteristics of the signal – for time-varying signals the available samples some distance in time away from the missing samples may not be relevant because the signal characteristics may have completely changed. This is particularly important in interpolation of a large sequence of samples.
- (4) The length of the missing samples – in general, interpolation quality decreases with increasing length of the missing samples.
- (5) Finally, interpolation depends on the optimal use of the data and the efficiency of the interpolator.

The classical approach to interpolation is to construct a polynomial interpolator function that passes through the known samples. We continue this chapter with a study of the general form of polynomial interpolation, and consider Lagrange, Newton, Hermite and cubic spline interpolators. Polynomial interpolators are *not* optimal or well suited to making efficient use of a relatively large number of known samples, or interpolating a relatively large segment of missing samples.

In Section 10.3, we study several statistical digital signal processing methods for interpolation of a sequence of missing samples. These include model-based methods, which

are well suited for interpolation of small to medium-sized gaps of missing samples. We also consider frequency–time interpolation methods, and interpolation through waveform substitution, which have the ability to replace relatively large gaps of missing samples.

10.2 POLYNOMIAL INTERPOLATION

The classical approach to interpolation is to construct a polynomial interpolator that passes through the known samples. Polynomial interpolators may be formulated in various forms, such as power series, Lagrange interpolation and Newton interpolation. These various forms are mathematically equivalent and can be transformed from one into another. Suppose the data consists of $N + 1$ samples $\{x(t_0), x(t_1), \dots, x(t_N)\}$, where $x(t_n)$ denotes the amplitude of the signal $x(t)$ at time t_n . The polynomial of order N that passes through the $N + 1$ known samples is unique (Figure 10.5) and may be written in power series form as

$$\hat{x}(t) = p_N(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + \dots + a_Nt^N \tag{10.13}$$

where $P_N(t)$ is a polynomial of order N , and the a_k are the polynomial coefficients. From Equation (10.13), and a set of $N + 1$ known samples, a system of $N + 1$ linear equations with $N + 1$ unknown coefficients can be formulated as

$$\begin{aligned} x(t_0) &= a_0 + a_1t_0 + a_2t_0^2 + a_3t_0^3 + \dots + a_Nt_0^N \\ x(t_1) &= a_0 + a_1t_1 + a_2t_1^2 + a_3t_1^3 + \dots + a_Nt_1^N \\ &\vdots \quad \vdots \quad \ddots \\ x(t_N) &= a_0 + a_1t_N + a_2t_N^2 + a_3t_N^3 + \dots + a_Nt_N^N \end{aligned} \tag{10.14}$$

From Equation (10.14) the polynomial coefficients are given by

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 & \dots & t_0^N \\ 1 & t_1 & t_1^2 & t_1^3 & \dots & t_1^N \\ 1 & t_2 & t_2^2 & t_2^3 & \dots & t_2^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 & \dots & t_N^N \end{pmatrix}^{-1} \begin{pmatrix} x(t_0) \\ x(t_1) \\ x(t_2) \\ \vdots \\ x(t_N) \end{pmatrix} \tag{10.15}$$

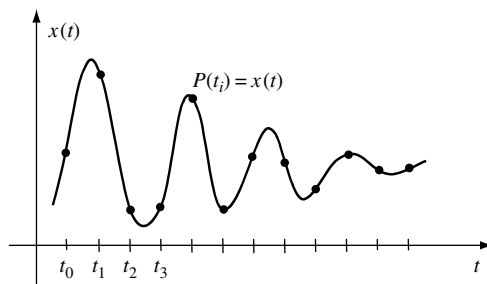


Figure 10.5 Illustration of an interpolation curve through a number of samples.

The matrix in Equation (10.15) is called a Vandermonde matrix. For a large number of samples, N , the Vandermonde matrix becomes large and ill-conditioned. An ill-conditioned matrix is sensitive to small computational errors, such as quantisation errors, and can easily produce inaccurate results. There are alternative methods of implementation of the polynomial interpolator that are simpler to program and/or better structured, such as Lagrange and Newton methods. However, it must be noted that these variants of the polynomial interpolation also become ill-conditioned for a large number of samples, N .

10.2.1 LAGRANGE POLYNOMIAL INTERPOLATION

To introduce the Lagrange interpolation, consider a line interpolator passing through two points $x(t_0)$ and $x(t_1)$:

$$\hat{x}(t) = p_1(t) = x(t_0) + \underbrace{\frac{x(t_1) - x(t_0)}{t_1 - t_0}}_{\text{Line slope}}(t - t_0) \tag{10.16}$$

The line Equation (10.16) may be rearranged and expressed as

$$p_1(t) = \frac{t - t_1}{t_0 - t_1} x(t_0) + \frac{t - t_0}{t_1 - t_0} x(t_1) \tag{10.17}$$

Equation (10.17) is in the form of a Lagrange polynomial. Note that the Lagrange form of a line interpolator is composed of the weighted combination of two lines, as illustrated in Figure 10.6.

In general, the Lagrange polynomial, of order N , passing through $N + 1$ samples $\{x(t_0), x(t_1), \dots, x(t_N)\}$ is given by the polynomial equation

$$P_N(t) = L_0(t)x(t_0) + L_1(t)x(t_1) + \dots + L_N(t)x(t_N) \tag{10.18}$$

where each Lagrange coefficient $L_N(t)$ is itself a polynomial of degree N given by

$$L_i(t) = \frac{(t - t_0) \cdots (t - t_{i-1})(t - t_{i+1}) \cdots (t - t_N)}{(t_i - t_0) \cdots (t_i - t_{i-1})(t_i - t_{i+1}) \cdots (t_i - t_N)} = \prod_{\substack{n=0 \\ n \neq i}}^N \frac{t - t_n}{t_i - t_n} \tag{10.19}$$

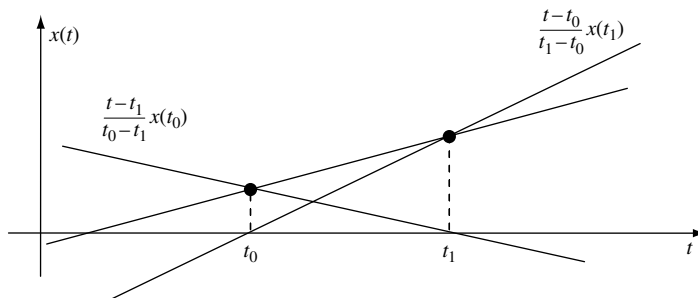


Figure 10.6 The Lagrange line interpolator passing through $x(t_0)$ and $x(t_1)$ described in terms of the combination of two lines: one passing through $[x(t_0), t_1]$ and the other through $[x(t_1), t_0]$.

Note that the i th Lagrange polynomial coefficient $L_i(t)$ becomes unity at the i th known sample point [i.e. $L_i(t_i) = 1$], and zero at every other known sample [i.e. $L_i(t_j) = 0, i \neq j$]. Therefore $P_N(t_i) = L_i(t_i)x(t_i) = x(t_i)$, and the polynomial passes through the known data points as required.

The main drawbacks of the Lagrange interpolation method are as follows:

- (1) the computational complexity is large.
- (2) the coefficients of a polynomial of order N cannot be used in the calculations of the coefficients of a higher-order polynomial;
- (3) the evaluation of the interpolation error is difficult.

The Newton polynomial, introduced in the next section, overcomes some of these difficulties.

10.2.2 NEWTON POLYNOMIAL INTERPOLATION

Newton polynomials have a recursive structure, such that a polynomial of order N can be constructed by extension of a polynomial of order $N - 1$ as follows:

$$\begin{aligned}
 p_0(t) &= a_0 && \text{(d.c. value)} \\
 p_1(t) &= a_0 + a_1(t - t_0) && \text{(ramp)} \\
 &= p_0(t) + a_1(t - t_0) \\
 p_2(t) &= \underbrace{a_0 + a_1(t - t_0)} + a_2(t - t_0)(t - t_1) && \text{(quadratic)} \\
 &= p_1(t) + a_2(t - t_0)(t - t_1) \\
 p_3(t) &= \underbrace{a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1)} + a_3(t - t_0)(t - t_1)(t - t_2) && \text{(cubic)} \\
 &= p_2(t) + a_3(t - t_0)(t - t_1)(t - t_2)
 \end{aligned} \tag{10.20}$$

and in general the recursive, *order update*, form of a Newton polynomial can be formulated as

$$p_N(t) = p_{N-1}(t) + a_N(t - t_0)(t - t_1) \cdots (t - t_{N-1}) \tag{10.21}$$

For a sequence of $N + 1$ samples $\{x(t_0), x(t_1), \dots, x(t_N)\}$, the polynomial coefficients are obtained using the constraint $p_N(t_i) = x(t_i)$ as follows: to solve for the coefficient a_0 , equate the polynomial Equation (10.21) at $t = t_0$ to $x(t_0)$:

$$p_N(t_0) = p_0(t_0) = x(t_0) = a_0 \tag{10.22}$$

To solve for the coefficient a_1 , the first-order polynomial $p_1(t)$ is evaluated at $t = t_1$:

$$p_1(t_1) = x(t_1) = a_0 + a_1(t_1 - t_0) = x(t_0) + a_1(t_1 - t_0) \tag{10.23}$$

from which

$$a_1 = \frac{x(t_1) - x(t_0)}{t_1 - t_0} \tag{10.24}$$

Note that the coefficient a_1 is the slope of the line passing through the points $[x(t_0), x(t_1)]$. To solve for the coefficient a_2 , the second-order polynomial $p_2(t)$ is evaluated at $t = t_2$:

$$p_2(t_2) = x(t_2) = a_0 + a_1(t_2 - t_0) + a_2(t_2 - t_0)(t_2 - t_1) \quad (10.25)$$

Substituting a_0 and a_1 from Equations (10.22) and (10.24) in Equation (10.25), we obtain

$$a_2 = \left[\frac{x(t_2) - x(t_1)}{t_2 - t_1} - \frac{x(t_1) - x(t_0)}{t_1 - t_0} \right] / (t_2 - t_0) \quad (10.26)$$

Each term in the square brackets of Equation (10.26) is a slope term, and the coefficient a_2 is the slope of the slope. To formulate a solution for the higher-order coefficients, we need to introduce the concept of *divided differences*. Each of the two ratios in the square brackets of Equation (10.26) is a so-called ‘divided difference’. The divided difference between two points t_i and t_{i-1} is defined as

$$d_1(t_{i-1}, t_i) = \frac{x(t_i) - x(t_{i-1})}{t_i - t_{i-1}} \quad (10.27)$$

The divided difference between two points may be interpreted as the average difference or the slope of the line passing through the two points. The second-order divided difference (i.e. the divided difference of the divided difference) over three points, t_{i-2} , t_{i-1} and t_i , is given by

$$d_2(t_{i-2}, t_i) = \frac{d_1(t_{i-1}, t_i) - d_1(t_{i-2}, t_{i-1})}{t_i - t_{i-2}} \quad (10.28)$$

and the third-order divided difference is

$$d_3(t_{i-3}, t_i) = \frac{d_2(t_{i-2}, t_i) - d_2(t_{i-3}, t_{i-1})}{t_i - t_{i-3}} \quad (10.29)$$

and so on. In general, the j th order divided difference can be formulated in terms of the divided differences of order $j - 1$, in an order-update equation given as

$$d_j(t_{i-j}, t_i) = \frac{d_{j-1}(t_{i-j+1}, t_i) - d_{j-1}(t_{i-j}, t_{i-1})}{t_i - t_{i-j}} \quad (10.30)$$

Note that $a_1 = d_1(t_0, t_1)$, $a_2 = d_2(t_0, t_2)$ and $a_3 = d_3(t_0, t_3)$, and in general the Newton polynomial coefficients are obtained from the divided differences using the relation

$$a_i = d_i(t_0, t_i) \quad (10.31)$$

A main advantage of the Newton polynomial is its computational efficiency, in that a polynomial of order $N - 1$ can be easily extended to a higher-order polynomial of order N . This is a useful property in the selection of the best polynomial order for a given set of data.

10.2.3 HERMITE POLYNOMIAL INTERPOLATION

Hermite polynomials are formulated to fit not only to the signal samples, but also to the derivatives of the signal as well. Suppose the data consists of $N + 1$ samples and assume that all the derivatives up to the M th order derivative are available. Let the data set, i.e. the signal samples and the derivatives, be denoted as $[x(t_i), x'(t_i), x''(t_i), \dots, x^{(M)}(t_i), i = 0, \dots, N]$. There are altogether $K = (N + 1)(M + 1)$ data points and a polynomial of order $K - 1$ can be fitted to the data as

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \dots + a_{K-1} t^{K-1} \quad (10.32)$$

To obtain the polynomial coefficients, we substitute the given samples in the polynomial and its M derivatives as

$$\begin{aligned} p(t_i) &= x(t_i) \\ p'(t_i) &= x'(t_i) \\ p''(t_i) &= x''(t_i) \\ &\vdots \\ &\vdots \\ p^{(M)}(t_i) &= x^{(M)}(t_i), \quad i = 0, 1, \dots, N \end{aligned} \quad (10.33)$$

In all, there are $K = (M + 1)(N + 1)$ equations in Equation (10.33), and these can be used to calculate the coefficients of the polynomial Equation (10.32). In theory, the constraint that the polynomial must also fit the derivatives should result in a better interpolating polynomial that passes through the sampled points and is also consistent with the known underlying dynamics (i.e. the derivatives) of the curve. However, even for moderate values of N and M , Equation (10.33) becomes too large for most practical purposes.

10.2.4 CUBIC SPLINE INTERPOLATION

A polynomial interpolator of order N is constrained to pass through $N + 1$ known samples, and can have $N - 1$ maxima and minima. In general, the interpolation error increases rapidly with the increasing polynomial order, as the interpolating curve has to wiggle through the $N + 1$ samples. When a large number of samples are to be fitted with a smooth curve, it may be better to divide the signal into a number of smaller intervals, and to fit a low-order interpolating polynomial to each small interval. Care must be taken to ensure that the polynomial curves are continuous at the endpoints of each interval. In cubic spline interpolation, a cubic polynomial is fitted to each interval between two samples. A cubic polynomial has the form

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (10.34)$$

A cubic polynomial has four coefficients, and needs four conditions for the determination of a unique set of coefficients. For each interval, two conditions are set by the samples at the endpoints of the interval. Two further conditions are met by the constraints that the

first derivatives of the polynomial should be continuous across each of the two endpoints. Consider an interval $t_i \leq t \leq t_{i+1}$ of length $T_i = t_{i+1} - t_i$, as shown in Figure 10.7. Using a local coordinate $\tau = t - t_i$, the cubic polynomial becomes

$$p(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 \quad (10.35)$$

At $\tau = 0$, we obtain the first coefficient a_0 as

$$a_0 = p(\tau = 0) = x(t_i) \quad (10.36)$$

The second derivative of $p(\tau)$ is given by

$$p''(\tau) = 2a_2 + 6a_3\tau \quad (10.37)$$

Evaluation of the second derivative at $\tau = 0$ (i.e. $t = t_i$) gives the coefficient a_2

$$a_2 = \frac{p''(\tau = 0)}{2} = \frac{p''_i}{2} \quad (10.38)$$

Similarly, evaluating the second derivative at the point t_{i+1} (i.e. $\tau = T_i$) yields the fourth coefficient,

$$a_3 = \frac{p''_{i+1} - p''_i}{6T_i} \quad (10.39)$$

Now to obtain the coefficient a_1 , we evaluate $p(\tau)$ at $\tau = T_i$:

$$p(\tau = T_i) = a_0 + a_1T_i + a_2T_i^2 + a_3T_i^3 = x(t_{i+1}) \quad (10.40)$$

and substitute a_0 , a_2 and a_3 from Equations (10.36), (10.38) and (10.39) in (10.40) to obtain

$$a_1 = \frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p''_{i+1} + 2p''_i}{6} T_i \quad (10.41)$$

The cubic polynomial can now be written as

$$p(\tau) = x(t_i) + \left[\frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p''_{i+1} + 2p''_i}{6} T_i \right] \tau + \frac{p''_i}{2} \tau^2 + \frac{p''_{i+1} - p''_i}{6T_i} \tau^3 \quad (10.42)$$

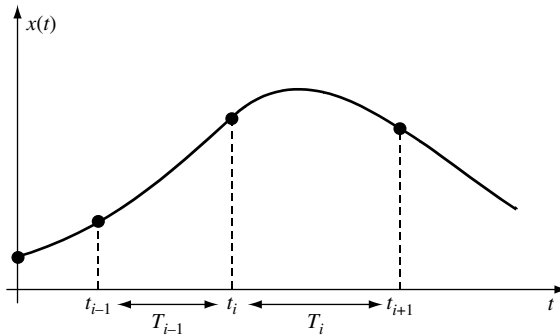


Figure 10.7 Illustration of cubic spline interpolation.

To determine the coefficients of the polynomial in Equation (10.42), we need the second derivatives p_i'' and p_{i+1}'' . These are obtained from the constraint that the first derivatives of the curves at the endpoints of each interval must be continuous. From Equation (10.42), the first derivatives of $p(\tau)$ evaluated at the endpoints t_i and t_{i+1} are

$$p'_i = p'(\tau = 0) = -\frac{T_i}{6} [p_{i+1}'' + 2p_i''] + \frac{1}{T_i} [x(t_{i+1}) - x(t_i)] \quad (10.43)$$

$$p'_{i+1} = p'(\tau = T_i) = \frac{T_i}{6} [2p_{i+1}'' + p_i''] + \frac{1}{T_i} [x(t_{i+1}) - x(t_i)] \quad (10.44)$$

Similarly, for the preceding interval, $t_{i-1} < t < t_i$, the first derivative of the cubic spline curve evaluated at $\tau = t_i$ is given by

$$p'_i = p'(\tau = t_i) = \frac{T_{i-1}}{6} [2p_i'' + p_{i-1}''] + \frac{1}{T_{i-1}} [x(t_i) - x(t_{i-1})] \quad (10.45)$$

For continuity of the first derivative at t_i , p'_i at the end of the interval (t_{i-1}, t_i) must be equal to the p'_i at the start of the interval (t_i, t_{i+1}) . Equating the right-hand sides of Equations (10.43) and (10.45) and repeating this exercise yields

$$T_{i-1}p_{i-1}'' + 2(T_{i-1} + T_i)p_i'' + T_i p_{i+1}'' = 6 \left[\frac{1}{T_{i-1}} x(t_{i-1}) - \left(\frac{1}{T_{i-1}} + \frac{1}{T_i} \right) x(t_i) + \frac{1}{T_i} x(t_{i+1}) \right] \\ i = 1, 2, \dots, N-1 \quad (10.46)$$

In Equation (10.46), there are $N-1$ equations in $N+1$ unknowns, p_i'' . For a unique solution we need to specify the second derivatives at the points t_0 and t_N . This can be done in two ways: (a) setting the second derivatives at the endpoints t_0 and t_N (i.e. p_0'' and p_N''), to zero; or (b) extrapolating the derivatives from the inside data.

10.3 MODEL-BASED INTERPOLATION

The statistical signal processing approach to interpolation of a sequence of lost samples is based on the utilisation of a predictive and/or a probabilistic model of the signal. In this section, we study the maximum *a posteriori* interpolation, an autoregressive model-based interpolation, a frequency-time interpolation method and interpolation through searching a signal record for the best replacement.

Figures 10.8 and 10.9 illustrate the problem of interpolation of a sequence of lost samples. It is assumed that we have a signal record of N samples, and that within this record a segment of M samples, starting at time k , $\mathbf{x}_{\text{Uk}} = \{x(k), x(k+1), \dots, x(k+M-1)\}$, is missing. The objective is to make an optimal estimate of the missing segment \mathbf{x}_{Uk} , using the remaining $N-k$ samples \mathbf{x}_{Kn} and a model of the signal process. An N -sample signal vector \mathbf{x} , composed of M unknown samples and $N-M$ known samples, can be written as

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_{\text{Kn}_1} \\ \mathbf{x}_{\text{U}} \\ \mathbf{x}_{\text{Kn}_2} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{\text{Kn}_1} \\ \mathbf{0} \\ \mathbf{x}_{\text{Kn}_2} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{x}_{\text{Uk}} \\ \mathbf{0} \end{pmatrix} = \mathbf{K}\mathbf{x}_{\text{Kn}} + \mathbf{U}\mathbf{x}_{\text{Uk}} \quad (10.47)$$

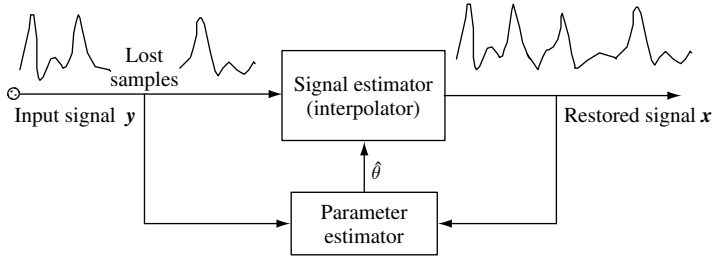


Figure 10.8 Illustration of a model-based iterative signal interpolation system.

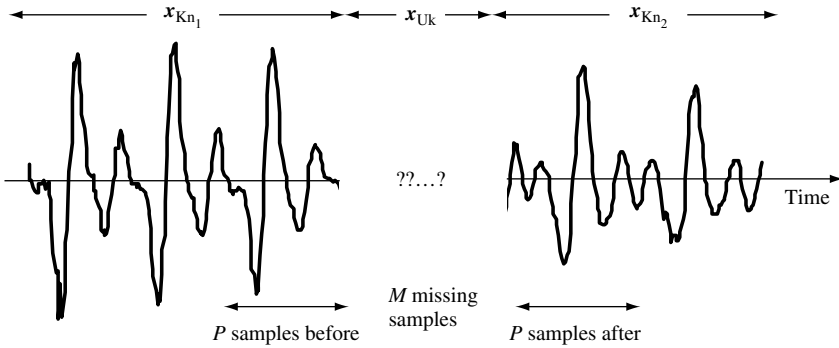


Figure 10.9 A signal with M missing samples and $N - M$ known samples. On each side of the missing segment, P samples are used to interpolate the segment.

where the vector $\mathbf{x}_{K_n} = [\mathbf{x}_{K_{n1}} \mathbf{x}_{K_{n2}}]^T$ is composed of the known samples, and the vector \mathbf{x}_{U_k} is composed of the unknown samples, as illustrated in Figure 10.8. The matrices \mathbf{K} and \mathbf{U} in Equation (10.47) are rearrangement matrices that assemble the vector \mathbf{x} from \mathbf{x}_{K_n} and \mathbf{x}_{U_k} .

10.3.1 MAXIMUM A POSTERIORI INTERPOLATION

The posterior pdf of an unknown signal segment, \mathbf{x}_{U_k} , given a number of neighbouring samples, \mathbf{x}_{K_n} , can be expressed using Bayes’s rule as

$$\begin{aligned}
 f_X(\mathbf{x}_{U_k} | \mathbf{x}_{K_n}) &= \frac{f_X(\mathbf{x}_{K_n}, \mathbf{x}_{U_k})}{f_X(\mathbf{x}_{K_n})} \\
 &= \frac{f_X(\mathbf{x} = \mathbf{K}\mathbf{x}_{K_n} + \mathbf{U}\mathbf{x}_{U_k})}{f_X(\mathbf{x}_{K_n})}
 \end{aligned}
 \tag{10.48}$$

In Equation (10.48), for a given sequence of samples \mathbf{x}_{K_n} , $f_X(\mathbf{x}_{K_n})$ is a constant. Therefore the estimate that maximises the posterior pdf, i.e. the MAP estimate, is given by

$$\hat{\mathbf{x}}_{U_k}^{\text{MAP}} = \arg \max_{\mathbf{x}_{U_k}} f_X(\mathbf{K}\mathbf{x}_{K_n} + \mathbf{U}\mathbf{x}_{U_k})
 \tag{10.49}$$

Example 10.2: MAP Interpolation of a Gaussian Signal

Assume that an observation signal $\mathbf{x} = \mathbf{K}\mathbf{x}_{\text{Kn}} + \mathbf{U}\mathbf{x}_{\text{Uk}}$, from a zero-mean Gaussian process, is composed of a sequence of M missing samples \mathbf{x}_{Uk} and $N - M$ known neighbouring samples as in Equation (10.47). The pdf of the signal \mathbf{x} is given by

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\Sigma_{\text{xx}}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T \Sigma_{\text{xx}}^{-1} \mathbf{x}\right) \quad (10.50)$$

where Σ_{xx} is the covariance matrix of the Gaussian vector process \mathbf{x} . Substitution of Equation (10.50) in Equation (10.48) yields the conditional pdf of the unknown signal \mathbf{x}_{Uk} given a number of samples \mathbf{x}_{Kn} :

$$f_X(\mathbf{x}_{\text{Uk}} | \mathbf{x}_{\text{Kn}}) = \frac{1}{f_X(\mathbf{x}_{\text{Kn}})} \frac{1}{(2\pi)^{N/2} |\Sigma_{\text{xx}}|^{1/2}} \times \exp\left[-\frac{1}{2} (\mathbf{K} \mathbf{x}_{\text{Kn}} + \mathbf{U} \mathbf{x}_{\text{Uk}})^T \Sigma_{\text{xx}}^{-1} (\mathbf{K} \mathbf{x}_{\text{Kn}} + \mathbf{U} \mathbf{x}_{\text{Uk}})\right] \quad (10.51)$$

The MAP signal estimate, obtained by setting the derivative of the log-likelihood function $\ln f_X(\mathbf{x} | \mathbf{x}_{\text{Kn}})$ of Equation (10.51) with respect to \mathbf{x}_{Uk} to zero, is given by

$$\mathbf{x}_{\text{Uk}} = -(\mathbf{U}^T \Sigma_{\text{xx}}^{-1} \mathbf{U})^{-1} \mathbf{U}^T \Sigma_{\text{xx}}^{-1} \mathbf{K} \mathbf{x}_{\text{Kn}} \quad (10.52)$$

An example of MAP interpolation is shown in Figure 10.10.

10.3.2 LEAST SQUARE ERROR AUTOREGRESSIVE INTERPOLATION

In this section, we describe interpolation based on an autoregressive model of the signal process. The term ‘autoregressive model’ is an alternative terminology for the linear predictive models considered in Chapter 8. In this section, the terms ‘linear predictive model’ and ‘autoregressive model’ are used interchangeably. The AR interpolation algorithm is a two-stage process: in the first stage, the AR model coefficients are estimated from the incomplete signal, and in the second stage the estimates of the model coefficients are used

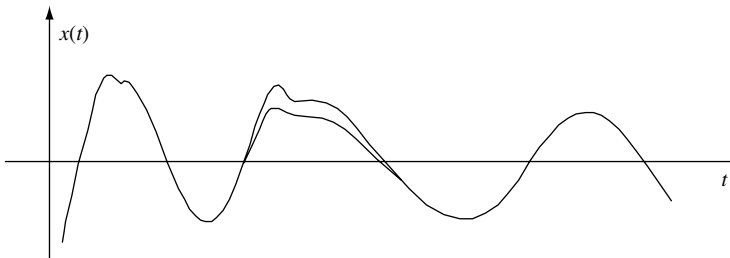


Figure 10.10 Illustration of MAP interpolation of a segment of 20 samples.

to interpolate the missing samples. For high-quality interpolation, the estimation algorithm should utilise all the correlation structures of the signal process, including periodic or pitch period structures. In Section 10.3.4, the AR interpolation method is extended to include pitch-period correlations.

10.3.3 INTERPOLATION BASED ON A SHORT-TERM PREDICTION MODEL

An autoregressive, or linear predictive, signal $x(m)$ is described as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \tag{10.53}$$

where $x(m)$ is the AR signal, a_k are the model coefficients and $e(m)$ is a zero mean excitation signal. The excitation may be a random signal, a quasiperiodic impulse train, or a mixture of the two. The AR coefficients, a_k , model the correlation structure or equivalently the spectral patterns of the signal.

Assume that we have a signal record of N samples and that within this record a segment of M samples, starting from the sample k , $\mathbf{x}_{\text{Uk}} = \{x(k), \dots, x(k+M-1)\}$, is missing. The objective is to estimate the missing samples \mathbf{x}_{Uk} , using the remaining $N-k$ samples and an AR model of the signal. Figure 10.8 illustrates the interpolation problem. For this signal record of N samples, the AR Equation (10.53) can be expanded to form the following matrix equation:

$$\begin{pmatrix} e^{(P)} \\ e^{(P+1)} \\ \vdots \\ e^{(k-1)} \\ e^{(k)} \\ e^{(k+1)} \\ e^{(k+2)} \\ \vdots \\ e^{(k+M+P-2)} \\ e^{(k+M+P-1)} \\ e^{(k+M+P)} \\ e^{(k+M+P+1)} \\ \vdots \\ e^{(N-1)} \end{pmatrix} = \begin{pmatrix} x^{(P)} \\ x^{(P+1)} \\ \vdots \\ x^{(k-1)} \\ \mathbf{x}_{\text{Uk}}(k) \\ \mathbf{x}_{\text{Uk}}(k+1) \\ \mathbf{x}_{\text{Uk}}(k+2) \\ \vdots \\ x^{(k+M+P-2)} \\ x^{(k+M+P-1)} \\ x^{(k+M+P)} \\ x^{(k+M+P+1)} \\ \vdots \\ x^{(N-1)} \end{pmatrix} - \begin{pmatrix} x^{(P-1)} & x^{(P-2)} & \dots & x^{(0)} \\ x^{(P)} & x^{(P-1)} & \dots & x^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k-2)} & x^{(k-3)} & \dots & x^{(k-P-1)} \\ x^{(k-1)} & x^{(k-2)} & \dots & x^{(k-P)} \\ \mathbf{x}_{\text{Uk}}(k) & \mathbf{x}_{\text{Uk}}(k) & \dots & \mathbf{x}_{\text{Uk}}(k-P+1) \\ \mathbf{x}_{\text{Uk}}(k+1) & \mathbf{x}_{\text{Uk}}(k) & \dots & \mathbf{x}_{\text{Uk}}(k-P+2) \\ \vdots & \vdots & \ddots & \vdots \\ x^{(k+M+P-3)} & x^{(k+M+P-4)} & \dots & \mathbf{x}_{\text{Uk}}(k+M-2) \\ x^{(k+M+P-2)} & x^{(k+M+P-3)} & \dots & \mathbf{x}_{\text{Uk}}(k+M-1) \\ x^{(k+M+P-1)} & x^{(k+M+P-2)} & \dots & x^{(k+M)} \\ x^{(k+M+P)} & x^{(k+M+P-1)} & \dots & x^{(k+M+1)} \\ \dots & \dots & \ddots & \dots \\ x^{(N-2)} & x^{(N-3)} & \dots & x^{(N-P-1)} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{10.54}$$

where the subscript Uk denotes the unknown samples. Equation (10.54) can be rewritten in compact vector notation as

$$\mathbf{e}(\mathbf{x}_{\text{Uk}}, \mathbf{a}) = \mathbf{x} - \mathbf{X}\mathbf{a} \tag{10.55}$$

where the error vector $\mathbf{e}(\mathbf{x}_{\text{Uk}}, \mathbf{a})$ is expressed as a function of the unknown samples and the unknown model coefficient vector. In this section, the optimality criterion for the estimation of the model coefficient vector, \mathbf{a} , and the missing samples, \mathbf{x}_{Uk} , is the minimum mean square error given by the inner vector product

$$\mathbf{e}^T \mathbf{e}(\mathbf{x}_{\text{Uk}}, \mathbf{a}) = \mathbf{x}^T \mathbf{x} + \mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a} - 2\mathbf{a}^T \mathbf{X}^T \mathbf{x} \tag{10.56}$$

The squared error function in Equation (10.56) involves nonlinear unknown terms of fourth order, $\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a}$, and cubic order, $\mathbf{a}^T \mathbf{X}^T \mathbf{x}$. The least square error formulation, obtained by differentiating $\mathbf{e}^T \mathbf{e}(\mathbf{x}_{\text{Uk}}, \mathbf{a})$, with respect to the vectors \mathbf{a} or \mathbf{x}_{Uk} , results in a set of nonlinear equations of cubic order whose solution is nontrivial. A suboptimal, but practical and mathematically tractable, approach is to solve for the missing samples and the unknown model coefficients in two separate stages. This is an instance of the general estimate–maximise algorithm, and is similar to the linear-predictive model-based restoration considered in Section 6.7. In the first stage of the solution, Equation (10.54) is *linearised* by either assuming that the missing samples have zero values or discarding the set of equations in Equation (10.54), between the two dashed lines, that involve the unknown signal samples. The linearised equations are used to solve for the AR model coefficient vector \mathbf{a} by forming the equation

$$\hat{\mathbf{a}} = (\mathbf{X}_{\text{Kn}}^T \mathbf{X}_{\text{Kn}})^{-1} (\mathbf{X}_{\text{Kn}}^T \mathbf{x}_{\text{Kn}}) \tag{10.57}$$

where the vector is an estimate of the model coefficients, obtained from the available signal samples.

The second stage of the solution involves the estimation of the unknown signal samples, \mathbf{x}_{Uk} . For an AR model of order P , and an unknown signal segment of length M , there are $2M + P$ nonlinear equations in Equation (10.54) that involve the unknown samples; these are

$$\begin{pmatrix} \mathbf{e}(k) \\ \mathbf{e}(k+1) \\ \mathbf{e}(k+2) \\ \vdots \\ \mathbf{e}(k+M+P-2) \\ \mathbf{e}(k+M+P-1) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{\text{Uk}}(k) \\ \mathbf{x}_{\text{Uk}}(k+1) \\ \mathbf{x}_{\text{Uk}}(k+2) \\ \vdots \\ \mathbf{x}(k+M+P-2) \\ \mathbf{x}(k+M+P-1) \end{pmatrix} - \begin{pmatrix} \mathbf{x}(k-1) & \mathbf{x}(k-2) & \dots & \mathbf{x}(k-p) \\ \mathbf{x}_{\text{Uk}}(k) & \mathbf{x}(k-1) & \dots & \mathbf{x}(k-p+1) \\ \mathbf{x}_{\text{Uk}}(k+1) & \mathbf{x}_{\text{Uk}}(k) & \dots & \mathbf{x}(k-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}(k+M+P-3) & \mathbf{x}_{\text{Uk}}(k+M+P-4) & \dots & \mathbf{x}_{\text{Uk}}(k+M-2) \\ \mathbf{x}(k+M+P-2) & \mathbf{x}_{\text{Uk}}(k+M+P-3) & \dots & \mathbf{x}_{\text{Uk}}(k+M-1) \end{pmatrix} \times \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \vdots \\ \mathbf{a}_{p-1} \\ \mathbf{a}_p \end{pmatrix} \tag{10.58}$$

The estimate of the predictor coefficient vector, obtained from the first stage of the solution, is substituted in Equation (10.58), so that the only remaining unknowns in Equation (10.58)

are the missing signal samples. Equation (10.58) may be partitioned and rearranged in vector notation in the following form:

$$\begin{pmatrix} e(k) \\ e(k+1) \\ e(k+2) \\ e(k+3) \\ e(k+4) \\ \vdots \\ e(k+P-1) \\ e(k+P) \\ e(k+P+1) \\ \vdots \\ e(k+m+P-2) \\ e(k+m+P-1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -a_1 & 1 & 0 & 0 & \dots & 0 \\ -a_2 & -a_1 & 1 & 0 & \dots & 0 \\ -a_3 & -a_2 & -a_1 & 1 & \dots & 0 \\ -a_4 & -a_3 & -a_2 & -a_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_p & -a_{p-1} & -a_{p-2} & -a_{p-3} & \dots & 0 \\ 0 & -a_p & -a_{p-1} & -a_{p-2} & \dots & 0 \\ 0 & 0 & -a_p & -a_{p-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -a_{p-1} \\ 0 & 0 & 0 & 0 & \dots & -a_p \end{pmatrix} \begin{pmatrix} \mathbf{x}_{\text{Uk}}(k) \\ \mathbf{x}_{\text{Uk}}(k+1) \\ \mathbf{x}_{\text{Uk}}(k+2) \\ \mathbf{x}_{\text{Uk}}(k+3) \\ \vdots \\ \mathbf{x}_{\text{Uk}}(k+M-1) \end{pmatrix} + \begin{pmatrix} -a_p & -a_{p-1} & -a_{p-2} & \dots & -a_1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & -a_p & -a_{p-1} & \dots & -a_2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -a_p & \dots & -a_3 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -a_p & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & -a_1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -a_2 & -a_1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -a_3 & -a_2 & -a_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \dots & -a_{p-1} & -a_{p-2} & -a_{p-3} & \dots & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}(k-P) \\ \mathbf{x}(k-P+1) \\ \mathbf{x}(k-P+2) \\ \vdots \\ \mathbf{x}(k-1) \\ 0 \\ \vdots \\ \mathbf{x}(k+M) \\ \mathbf{x}(k+M+1) \\ \mathbf{x}(k+M+2) \\ \vdots \\ \mathbf{x}(k+M+P-1) \end{pmatrix} \tag{10.59}$$

In Equation (10.59), the unknown and known samples are rearranged and grouped into two separate vectors. In a compact vector–matrix notation, Equation (10.58) can be written in the form

$$\mathbf{e} = \mathbf{A}_1 \mathbf{x}_{\text{Uk}} + \mathbf{A}_2 \mathbf{x}_{\text{Kn}} \tag{10.60}$$

where \mathbf{e} is the error vector, \mathbf{A}_1 is the first coefficient matrix, \mathbf{x}_{Uk} is the unknown signal vector being estimated, \mathbf{A}_2 is the second coefficient matrix and the vector \mathbf{x}_{Kn} consists of the *known* samples in the signal matrix and vectors of Equation (10.58). The total squared error is given by

$$\mathbf{e}^T \mathbf{e} = (\mathbf{A}_1 \mathbf{x}_{\text{Uk}} + \mathbf{A}_2 \mathbf{x}_{\text{Kn}})^T (\mathbf{A}_1 \mathbf{x}_{\text{Uk}} + \mathbf{A}_2 \mathbf{x}_{\text{Kn}}) \tag{10.61}$$

The least square AR (LSAR) interpolation is obtained by minimisation of the squared error function with respect to the unknown signal samples \mathbf{x}_{Uk} :

$$\frac{\partial \mathbf{e}^T \mathbf{e}}{\partial \mathbf{x}_{\text{Uk}}} = 2\mathbf{A}_1^T \mathbf{A}_1 \mathbf{x}_{\text{Uk}} + 2\mathbf{A}_1^T \mathbf{A}_2 \mathbf{x}_{\text{Kn}} = 0 \tag{10.62}$$

From Equation (10.62) we have

$$\hat{\mathbf{x}}_{\text{Uk}}^{\text{LSAR}} = -(\mathbf{A}_1^T \mathbf{A}_1)^{-1} (\mathbf{A}_1^T \mathbf{A}_2) \mathbf{x}_{\text{Kn}} \quad (10.63)$$

The solution in Equation (10.62) gives the $\hat{\mathbf{x}}_{\text{Uk}}^{\text{LSAR}}$ vector, which is the least square error estimate of the unknown data vector.

10.3.4 INTERPOLATION BASED ON LONG- AND SHORT-TERM CORRELATIONS

For the best results, a model-based interpolation algorithm should utilise all the correlation structures of the signal process, including any periodic structures. For example, the main correlation structures in a voiced speech signal are the short-term correlation due to the resonance of the vocal tract and the long-term correlation due to the quasiperiodic excitation pulses of the glottal cords. For voiced speech, interpolation based on the short-term correlation does not perform well if the missing samples coincide with an underlying quasiperiodic excitation pulse.

In this section, the AR interpolation is extended to include both long- and short-term correlations. For most audio signals, the short-term correlation of each sample with the immediately preceding samples decays exponentially with time, and can usually be modelled with an AR model of order 10–20. In order to include the pitch periodicities in the AR model of Equation (10.53), the model order must be greater than the pitch period. For speech signals, the pitch period is normally in the range 4–20 ms, equivalent to 40–200 samples at a sampling rate of 10 kHz. Implementation of an AR model of this order is not practical owing to stability problems and computational complexity.

A more practical AR model that includes the effects of the long-term correlations is illustrated in Figure 10.11. This modified AR model may be expressed by the following equation:

$$x(m) = \sum_{k=1}^P a_k x(m-k) + \sum_{k=-Q}^Q p_k x(m-T-k) + e(m) \quad (10.64)$$

The AR model of Equation (10.64) is composed of a *short-term predictor* $\sum a_k x(m-k)$ that models the contribution of the P immediate past samples, and a *long-term predictor* $\sum p_k x(m-T-k)$ that models the contribution of $2Q+1$ samples a pitch period away. The parameter T is the pitch period; it can be estimated from the autocorrelation function of $x(m)$ as the time difference between the peak of the autocorrelation, which is at the correlation lag zero, and the second largest peak, which should happen a pitch period away from the lag zero.

The AR model of Equation (10.64) is specified by the parameter vector $\mathbf{c} = [a_1, \dots, a_P, p_{-Q}, \dots, p_Q]$ and the pitch period T . Note that in Figure 10.11 the sample marked ‘?’ coincides with the onset of an excitation pulse. This sample is not very predictable from the P past samples, because they do not include a pulse event. The sample is more predictable from the $2Q+1$ samples a pitch period away, since they include the effects of

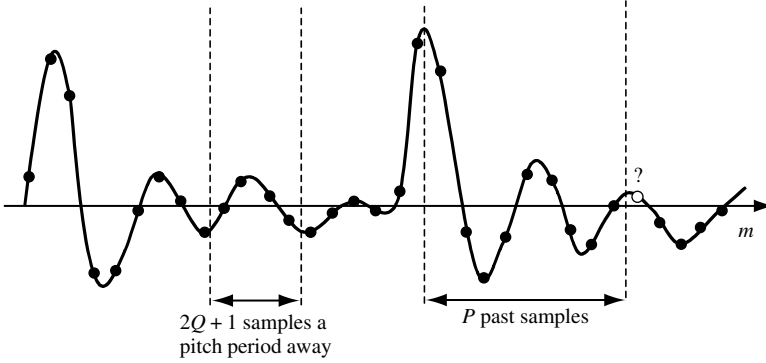


Figure 10.11 A quasiperiodic waveform. The sample marked ‘?’ is predicted using P immediate past samples and $2Q + 1$ samples a pitch period away.

a similar excitation pulse. The predictor coefficients are estimated (see Chapter 7) using the so-called normal equations:

$$c = R_{xx}^{-1}r_{xx} \tag{10.65}$$

where R_{xx} is the autocorrelation matrix of signal x and r_{xx} is the correlation vector. In expanded form, Equation (10.65) can be written as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \\ p-Q \\ p-Q+1 \\ \vdots \\ p+Q \end{pmatrix} = \begin{pmatrix} r(0) & r(1) & \dots & r(P-1) & r(T+Q-1) & r(T+Q) & \dots & r(T-Q-1) \\ r(1) & r(0) & \dots & r(P-2) & r(T+Q-2) & r(T+Q-1) & \dots & r(T+Q-2) \\ r(2) & r(1) & \dots & r(P-3) & r(T+Q-3) & r(T+Q-2) & \dots & r(T+Q-3) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \dots & r(0) & r(T+Q-P) & r(T+Q-P+1) & \dots & r(T+Q-P) \\ r(T+Q-1) & r(T+Q-2) & \dots & r(T+Q-P) & r(0) & r(1) & \dots & r(2Q) \\ r(T+Q) & r(T+Q-1) & \dots & r(T+Q-P+1) & r(1) & r(0) & \dots & r(2Q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(T-Q-1) & r(T-Q-2) & \dots & r(T-Q-P) & r(2Q) & r(2Q-1) & \dots & r(0) \end{pmatrix}^{-1} \times \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T+Q) \\ r(T+Q-1) \\ \vdots \\ r(T-Q) \end{pmatrix} \tag{10.66}$$

The modified AR model can be used for interpolation in the same way as the conventional AR model described in the previous section. Again, it is assumed that, within a data window of N speech samples, a segment of M samples commencing from the sample point k , $x_{Uk} = \{x(k), x(k+1), \dots, x(k+M-1)\}$ is missing. Figure 10.12 illustrates the interpolation problem. The missing samples are estimated using P samples in the immediate vicinity and

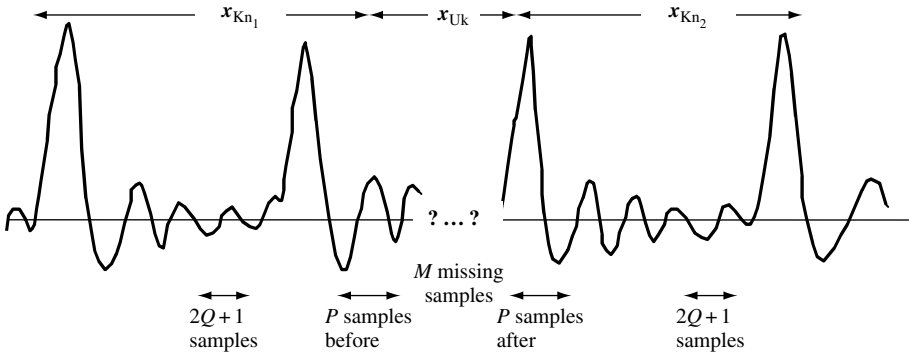


Figure 10.12 A signal with M missing samples. P immediate samples each side of the gap and $2Q + 1$ samples a pitch period away are used for interpolation.

$2Q + 1$ samples a pitch period away on each side of the missing signal. For the signal record of N samples, the modified AR equation (10.64) can be written in matrix form as

$$\begin{pmatrix} e(T+Q) \\ e(T+Q+1) \\ \vdots \\ e(k-1) \\ e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \\ e(k+M+P) \\ e(k+M+P+1) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(T+Q) \\ x(T+Q+1) \\ \vdots \\ x(k-1) \\ x_{Uk}(k) \\ x_{Uk}(k+1) \\ x_{Uk}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \\ x(k+M+P) \\ x(k+M+P+1) \\ \vdots \\ x(N-1) \end{pmatrix}$$

$$- \begin{pmatrix} x(T+Q-1) & \dots & x(T+Q-P) & & x(2Q) & \dots & x(0) \\ x(T+Q) & \dots & x(T+Q-P+1) & & x(2Q+1) & \dots & x(1) \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ x(k-2) & \dots & x(k-P-1) & & x(k-T+Q-1) & \dots & x(k-T-Q-1) \\ x(k-1) & \dots & x(k-P) & & x(k-T+Q) & \dots & x(k-T-Q) \\ x_{Uk}(k) & \dots & x(k-P+1) & & x(k-T+Q+1) & \dots & x(k-T-Q+1) \\ x_{Uk}(k+1) & \dots & x(k-P+2) & & x(k-T+Q+2) & \dots & x(k-T-Q+2) \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ x(k+M+P-3) & \dots & x_{Uk}(k+M-2) & & x(k+M+P-T+Q-2) & \dots & x(k+M+P-T-Q-2) \\ x(k+M+P-2) & \dots & x_{Uk}(k+M-1) & & x(k+M+P-T+Q-1) & \dots & x(k+M+P-T-Q-1) \\ x(k+M+P-1) & \dots & x(k+M) & & x(k+M+P-T+Q) & \dots & x(k+M+P-T-Q) \\ x(k+M+P) & \dots & x(k+M+1) & & x(k+M+P-T+Q+1) & \dots & x(k+M+P-T-Q+1) \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ x(N-2) & \dots & x(N-P-1) & & x(N-T+Q-1) & \dots & x(N-T-Q-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \\ p-q \\ \vdots \\ p+q \end{pmatrix} \tag{10.67}$$

where the subscript U_k denotes the unknown samples. In compact matrix notation, this set of equation can be written in the form

$$\mathbf{e}(\mathbf{x}_{U_k}, \mathbf{c}) = \mathbf{x} + \mathbf{X}\mathbf{c} \quad (10.68)$$

As in Section 10.3.2, the interpolation problem is solved in two stages:

- (1) in the first stage, the known samples on both sides of the missing signal are used to estimate the AR coefficient vector, \mathbf{c} ;
- (2) in the second stage, the AR coefficient estimates are substituted in Equation (10.68) so that the only unknowns are the data samples.

The solution follows the same steps as those described in Section 10.3.2.

10.3.5 LSAR INTERPOLATION ERROR

In this section, we discuss the effects of the signal characteristics, the model parameters and the number of unknown samples on the interpolation error. The interpolation error $v(m)$, defined as the difference between the original sample $x(m)$ and the interpolated sample $\hat{x}(m)$, is given by

$$v(m) = x(m) - \hat{x}(m) \quad (10.69)$$

A common measure of signal distortion is the mean square error distance, defined as

$$D(\mathbf{c}, M) = \frac{1}{M} \mathcal{E} \left\{ \sum_{m=0}^{M-1} [x(k+m) - \hat{x}(k+m)]^2 \right\} \quad (10.70)$$

where k is the beginning of an M -samples-long segment of missing signal, and $\mathcal{E}[\cdot]$ is the expectation operator. In Equation (10.70), the average distortion D is expressed as a function of the number of the unknown samples M , and also the model coefficient vector \mathbf{c} . In general, the quality of interpolation depends on the following factors:

- (1) *The signal correlation structure* – for deterministic signals such as sine waves, the theoretical interpolation error is zero. However information-bearing signals have a degree of randomness that makes perfect interpolation with zero error an impossible objective.
- (2) *The length of the missing segment* – the amount of information lost, and hence the interpolation error, increases with the number of missing samples. Within a sequence of missing samples the error is usually largest for the samples in the middle of the gap. The interpolation Equation (10.63) becomes increasingly ill-conditioned as the length of the missing samples increases.
- (3) *The nature of the excitation underlying the missing samples* – the LSAR interpolation cannot account for any random excitation underlying the missing samples. In particular, the interpolation quality suffers when the missing samples coincide with the onset of

an excitation pulse. In general, the least square error criterion causes the interpolator to underestimate the energy of the underlying excitation signal. The inclusion of long-term prediction and the use of quasiperiodic structure of signals improves the ability of the interpolator to restore the missing samples.

- (4) *AR model order and the method used for estimation of the AR coefficients* – the interpolation error depends on the AR model order. Usually a model order of two to three times the length of missing data sequence achieves good result.

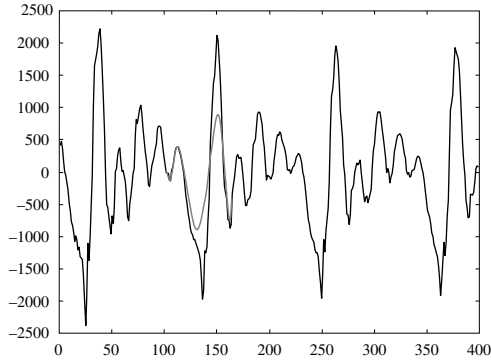
The interpolation error also depends on how well the AR parameters can be estimated from the incomplete data. In Equation (10.54), in the first stage of the solution, where the AR coefficients are estimated, two different approaches may be employed to linearise the system of equations. In the first approach, all equations between the dashed lines that involve nonlinear terms are discarded. This approach has the advantage that no assumption is made about the missing samples. In fact, from a signal-ensemble point of view, the effect of discarding some equations is equivalent to that of having a smaller signal record. In the second method, starting from an initial estimate of the unknown vector (such as $\mathbf{x}_{Uk} = \mathbf{0}$), Equation (10.54) is solved to obtain the AR parameters. The AR coefficients are then used in the second stage of the algorithm to estimate the unknown samples. These estimates may be improved in further iterations of the algorithm. The algorithm usually converges after one or two iterations.

Figures 10.13 and 10.14 show the results of application of the least square error AR interpolation method to speech signals. The interpolated speech segments were chosen to coincide with the onset of an excitation pulse. In these experimental cases the original signals are available for comparison. Each signal was interpolated by the AR model of Equation (10.53) and also by the extended AR model of Equation (10.64). The length of the conventional linear predictor model was set to 20. The modified linear AR model of Equation (10.64) has a prediction order of (20,7); that is, the short-term predictor has 20 coefficients and the long-term predictor has seven coefficients. The figures clearly demonstrate that the modified AR model that includes the long-term as well as the short-term correlation structures outperforms the conventional AR model.

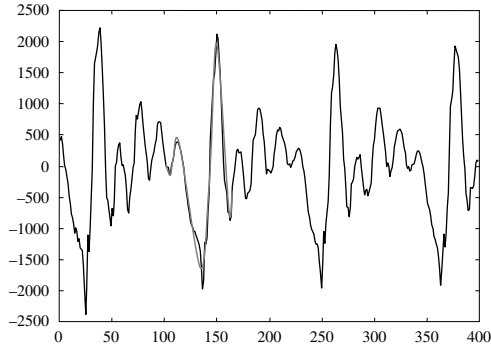
10.3.6 INTERPOLATION IN FREQUENCY–TIME DOMAIN

Time-domain, AR model-based interpolation methods are effective for the interpolation of a relatively short length of samples (say less than 100 samples at a 20 kHz sampling rate), but suffer severe performance degradations when used for interpolation of large sequence of samples. This is partly due to the numerical problems associated with the inversion of a large matrix, involved in the time-domain interpolation of a large number of samples, Equation (10.58).

Spectral–time representation provides a useful form for the interpolation of a large gap of missing samples. For example, through discrete Fourier transformation and spectral–time representation of a signal, the problem of interpolation of a gap of N samples in the time domain can be converted into the problem of interpolation of a gap of one sample, along the time, in each of N discrete frequency bins, as explained next.

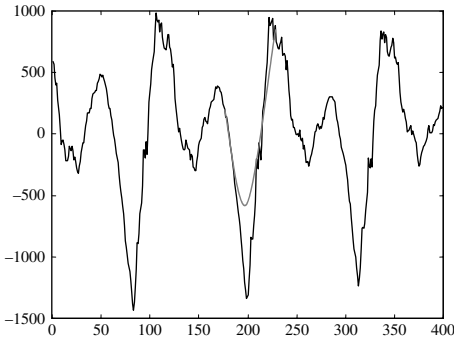


(a)

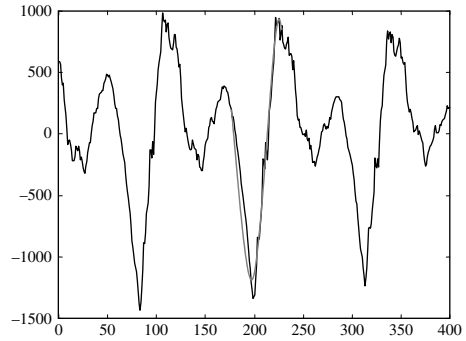


(b)

Figure 10.13 (a) A section of speech showing interpolation of 60 samples starting from the sample point 100. (b) Interpolation using short- and long-term correlations. Interpolated samples are shown by the dotted line.



(a)



(b)

Figure 10.14 (a) A section of speech showing interpolation of 50 samples starting from the sample point 175. (b) Interpolation using short- and long-term correlations. Interpolated samples are shown by the dotted line.

10.3.6.1 Spectral–Time Representation with STFT

A relatively simple and practical method for spectral–time representation of a signal is the short-time Fourier transform (STFT) method. To construct a two-dimensional STFT from a one-dimensional function of time, $x(m)$, the input signal is segmented into overlapping blocks of N samples, as illustrated in Figure 10.15. Each block is windowed, prior to discrete Fourier transformation, to reduce the spectral leakage due to the effects of discontinuities at the edges of the block. The frequency spectrum of the m th signal block is given by the discrete Fourier transform as

$$X(k, m) = \sum_{i=0}^{N-1} w(i)x[m(N - D) + i]e^{-j(2\pi/N)ik}, k = 0, \dots, N - 1 \tag{10.71}$$

where $X(k, m)$ is a spectral–time representation with discrete frame index m and discrete frequency index k , N is the number of samples in each block and D is the block overlap. In STFT, it is assumed that the signal frequency composition is time-invariant within the duration of each block, but it may vary across the blocks. In general, the k th spectral component of a signal has a time-varying character, i.e. it is ‘born’, evolves for some time, disappears, and then reappears with a different intensity and a different characteristics. Figure 10.16 illustrates a spectral–time signal with a missing block of samples. The aim of

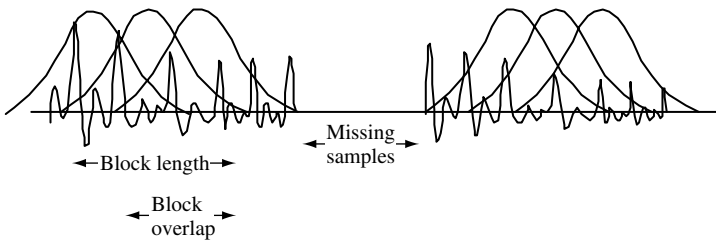


Figure 10.15 Illustration of segmentation of a signal (with a gap of missing samples) for spectral–time representation.

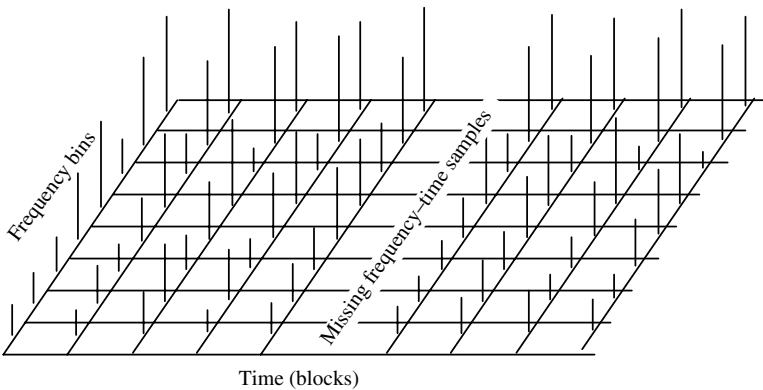


Figure 10.16 Spectral–time representation of a signal with a gap of missing samples.

interpolation is to fill in the signal gap such that, at the beginning and at the end of the gap, the continuity of both the magnitude and the phase of each frequency component of the signal is maintained. For most time-varying signals (such as speech), a low-order polynomial interpolator of the magnitude and the phase of the DFT components of the signal, making use of the few adjacent blocks on either side of the gap, would produce satisfactory results.

10.3.7 INTERPOLATION USING ADAPTIVE CODEBOOKS

In the LSAR interpolation method, described in Section 10.3.2, the signals are modelled as the output of an AR model excited by a random input. Given enough samples, the AR coefficients can be estimated with reasonable accuracy. However, the instantaneous values of the random excitation during the periods when the signal is missing cannot be recovered. This leads to a consistent underestimation of the amplitude and the energy of the interpolated samples. One solution to this problem is to use a zero-input signal model. Zero-input models are feedback oscillator systems that produce an output signal without requiring an input as illustrated in Figure 10.17.

The general form of the equation describing a digital nonlinear oscillator can be expressed as

$$x(m) = g_f [x(m - 1), x(m - 2), \dots, x(m - P)] \tag{10.72}$$

The mapping function $g_f(\cdot)$ may be a parametric or a nonparametric mapping. The model in Equation (10.72) can be considered as a nonlinear predictor, and the subscript f denotes forward prediction based on the past samples.

A parametric model of a nonlinear oscillator can be formulated using a Volterra filter model. However, in this section, we consider a nonparametric method for its ease of formulation and stable characteristics. Kubin and Kleijin (1994) have described a nonparametric oscillator based on a codebook model of the signal process.

In this method, each entry in the code book has $P + 1$ samples where the $(P + 1)$ th sample is intended as an output. Given P input samples, $\mathbf{x} = [x(m - 1), \dots, x(m - P)]$, the codebook output is the $(P + 1)$ th sample of the vector in the codebook whose first P samples have a minimum distance from the input signal \mathbf{x} . For a signal record of length N samples, a codebook of size $N - P$ vectors can be constructed by dividing the signal into overlapping segments of $P + 1$ samples with the successive segments having an overlap of P samples. Similarly a backward oscillator can be expressed as

$$x_b(m) = g_b [x(m + 1), x(m + 2), \dots, x(m + P)] \tag{10.73}$$

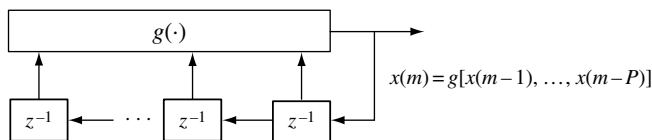


Figure 10.17 Configuration of a digital oscillator.

As in the case of a forward oscillator, the backward oscillator can be designed using a nonparametric method based on an adaptive codebook of the signal process. In this case each entry in the codebook has $P + 1$ samples where the first sample is intended as an output sample. Given P input samples, $\mathbf{x} = [x(m), \dots, x(m + P - 1)]$, the codebook output is the first sample of the codebook vector whose next P samples have a minimum distance from the input signal \mathbf{x} .

For interpolation of M missing samples, the outputs of the forward and backward nonlinear oscillators may be combined as

$$\hat{x}(k + m) = \left(\frac{M - 1 - m}{M - 1} \right) \hat{x}_f(k + m) + \left(\frac{m}{M - 1} \right) \hat{x}_b(k + m) \quad (10.74)$$

where it is assumed that the missing samples start at k .

10.3.8 INTERPOLATION THROUGH SIGNAL SUBSTITUTION

Audio signals often have a time-varying but quasiperiodic repetitive structure. Therefore most acoustic events in a signal record *reoccur* with some variations. This observation forms the basis for interpolation through pattern matching, where a missing segment of a signal is substituted by the best match from a signal record. Consider a relatively long signal record of N samples, with a gap of M missing samples at its centre. A section of the signal with the gap in the middle can be used to search for the best-match segment in the record. The missing samples are then substituted by the corresponding section of the best-match signal. This interpolation method is particularly useful when the length of the missing signal segment is large. For a given class of signals, we may be able to construct a library of patterns for use in waveform substitution (Bogner and Li, 1989).

10.4 SUMMARY

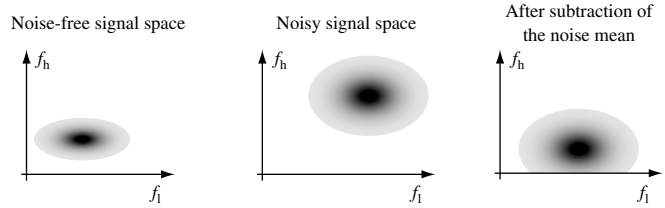
Interpolators, in their various forms, are used in most signal processing applications. The obvious example is the estimation of a sequence of missing samples. However, the use of an interpolator covers a much wider range of applications, from low-bit-rate speech coding to pattern recognition and decision-making systems. We started this chapter with a study of the ideal interpolation of a band-limited signal, and its applications in digital-to-analogue conversion and in multirate signal processing. In this chapter, various interpolation methods were categorised and studied in two different sections: one on polynomial interpolation, which is the more traditional numerical computing approach, and the other on statistical interpolation, which is the digital signal processing approach.

The general form of the polynomial interpolator was formulated and its special forms, Lagrange, Newton, Hermite and cubic spline interpolators, were considered. The polynomial methods are not equipped to make optimal use of the predictive and statistical structures of the signal, and are impractical for interpolation of a relatively large number of samples. A number of useful statistical interpolators were studied. These include maximum *a posteriori* interpolation, least square error AR interpolation, frequency–time interpolation, and an adaptive codebook interpolator. Model-based interpolation method based on an

autoregressive model is satisfactory for most audio applications so long as the length of the missing samples is not too large. For interpolation of a relatively large number of samples the time–frequency interpolation method and the adaptive code book method are more suitable.

BIBLIOGRAPHY

- Bogner R.E. and Li T. (1989) Pattern search prediction of speech. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-89*, Glasgow, pp. 180–183.
- Cohen L. (1989) Time–frequency distributions – a review. *Proc. IEEE*, **77**(7): 941–981.
- Crochiere R.E. and Rabiner L.R. (1981) Interpolation and decimation of digital signals – a tutorial review. *Proc. IEEE*, **69**: 300–331.
- Godsill S.J. (1993) The restoration of degraded audio signals. Ph.D. Thesis, Cambridge University, Cambridge.
- Godsill S.J. and Rayner P.J.W. (1993) Frequency domain interpolation of sampled signals. *Proceedings of IEEE International Conference on Speech and Signal Processing, ICASSP-93*, Minneapolis.
- Janssen A.J. and Vries L.B. (1984) Interpolation of band-limited discrete-time signals by minimising out-of-band energy. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-84*.
- Janssen A.J., Veldhuis R. and Vries L.B. (1984) Adaptive interpolation of discrete-time signals that can be modelled as autoregressive processes. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-34**(2): 317–330.
- Kay S.M. (1983) Some results in linear interpolation theory. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-31**: 746–749.
- Kay S.M. (1988) *Modern Spectral Estimation: Theory and Application*. Prentice-Hall, Englewood Cliffs, NJ.
- Kolmogorov A.N. (1939) Sur l' Interpolation et Extrapolation des Suites Stationnaires. *C. R. Acad. Sci.*, **208**: 2043–2045.
- Kubin G. and Kleijin W.B. (1994) Time-scale modification of speech based on a nonlinear oscillator model. *Proceedings of IEEE International Conference, Speech and Signal Processing, ICASSP-94*, Adelaide, pp. I-453–I-456.
- Lochart G.B. and Goodman D.J. (1986) Reconstruction of missing speech packets by waveform substitution. *Signal Processing 3: Theories and Applications*, pp. 357–360.
- Marks R.J. (1983) Restoring lost samples from an over-sampled band-limited signal. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-31**(2): 752–755.
- Marks R.J. (1991) *Introduction to Shannon Sampling and Interpolation Theory*. Springer, Berlin.
- Mathews J.H. (1992) *Numerical Methods for Mathematics, Science and Engineering*. Prentice-Hall, Englewood Cliffs, NJ.
- Musicus B.R. (1982) Iterative algorithms for optimal signal reconstruction and parameter identification given noisy and incomplete data. Ph.D. Thesis, MIT, Cambridge, MA.
- Nakamura S. (1991) *Applied Numerical Methods with Software*. Prentice-Hall, Englewood Cliffs, NJ.
- Platte H.J. and Rowedda V. (1985) A burst error concealment method for digital audio tape application. AES Preprint, 2201:1–16.
- Press W.H., Flannery B.P., Teukolsky S.A. and Vetterling W.T. (1992) *Numerical Recipes in C*, 2nd edn. Cambridge University Press, Cambridge.
- Schafer, R.W. and Rabiner, L.R. (1973) A digital signal processing approach to interpolation. *Proc. IEEE*, **61**: 692–702.
- Steele R. and Jayant N.S. (1980) Statistical block coding for DPCM-AQF speech. *IEEE Trans. Commun.*, **COM-28**(11): 1899–1907.
- Tong H. (1990) *Nonlinear Time Series. A Dynamical System Approach*. Oxford University Press, Oxford.
- Vaseghi S.V. (1988) Algorithms for restoration of gramophone records. Ph.D. Thesis, Cambridge University, Cambridge.
- Veldhuis R. (1990) *Restoration of Lost Samples in Digital Signals*. Prentice-Hall, Englewood Cliffs, NJ.
- Verhelst W. and Roelands M. (1993) An overlap-add technique based on waveform similarity (Wsola) for high quality time-scale modification of speech. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-93*, Adelaide, pp. II-554–II-557.
- Wiener N. (1949) *Extrapolation, Interpolation and Smoothing of Stationary Time Series With Engineering Applications*. MIT Press, Cambridge, MA.



Spectral Amplitude Estimation

11.1 Introduction	298
11.2 Spectral Subtraction	300
11.3 Bayesian MMSE Spectral Amplitude Estimation	312
11.4 Application to Speech Restoration and Recognition	315
11.5 Summary	315
Bibliography	316

Spectral amplitude estimation forms the basis of most signal restoration systems, such as for speech de-noising, where the phase distortion can be ignored. The simplest form of spectral amplitude estimation is spectral subtraction. This is a method for restoration of the power spectrum or the magnitude spectrum of a signal observed in additive noise, through subtraction of an estimate of the average noise spectrum from the noisy signal spectrum. The noise spectrum is usually estimated, and updated, from the periods when the signal is absent. For restoration of time-domain signals, an estimate of the instantaneous magnitude spectrum is combined with the phase of the noisy signal, and then transformed via an inverse discrete Fourier transform to the time domain. In terms of computational complexity, spectral subtraction is inexpensive. However, owing to random variations of noise, spectral subtraction can result in negative estimates of the short-time magnitude or power spectrum. The magnitude and power spectrum are non-negative variables, and any negative estimates of these variables should be mapped into non-negative values. This nonlinear rectification process distorts the distribution of the restored signal.

Bayesian spectral amplitude estimation methods offer substantial performance improvements on spectral subtraction by utilising the probability density functions of the signal and noise process. In particular, we consider the Bayesian spectral amplitude estimation using a minimum mean squared error cost function, as this method has received much attention in the quest for development of improved signal restoration algorithms.

11.1 INTRODUCTION

In multiple input signal restoration applications where, in addition to the noisy signal, the noise is accessible on a separate channel, it may be possible to retrieve the signal by subtracting an estimate of the noise from the noisy signal. For example, the adaptive noise canceller of Section 1.3.1 takes as the inputs the noise and the noisy signal, and as the outputs an estimate of the clean signal. However, in many applications, such as at the receiver of a noisy mobile phone, the only signal that is available is the noisy signal. In these single input applications, it is not possible to cancel out the random noise, but it may be possible to reduce the *average effects* of the noise on the spectral amplitude of the signal. The effect of additive noise on the spectral amplitude of a signal is an increase in the mean and the variance of the spectrum of the signal, as illustrated in Figure 11.1. The increase in the variance of the signal spectrum results from the random fluctuations of the noise.

Spectral amplitude estimation methods use a model of the distributions of the signal and the noise to provide an estimate of the amplitude spectrum of the clean signal. The effect of distortions of the phase of the signal spectrum is ignored. In its simplest form a spectral amplitude estimation method subtracts an estimate of the noise from the noisy signal. In more advanced forms, a Bayesian inference framework employs probability distributions of signal and noise to obtain more optimal results.

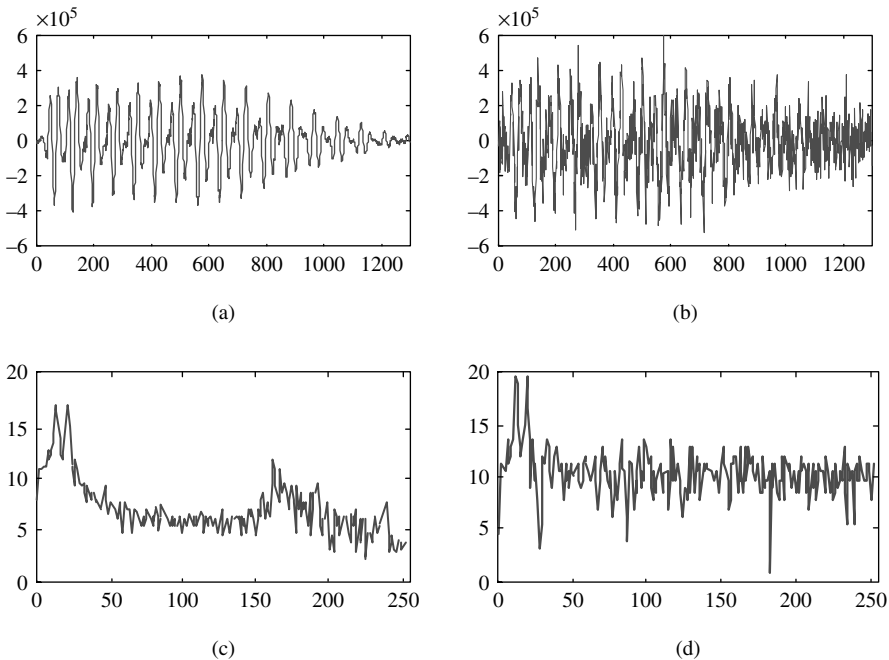


Figure 11.1 Illustrations of the effect of noise on a signal in the time and the frequency domains: (a) clean signal; (b) noisy signal; (c) spectrum of the clean signal; and (d) spectrum of the noisy signal.

11.1.1 SPECTRAL REPRESENTATION OF NOISY SIGNALS

Assuming that the noisy signal, $y(m)$, is modelled as the sum of the clean signal, $x(m)$, and the noise, $n(m)$, we have

$$y(m) = x(m) + n(m) \quad (11.1)$$

where the integer variable m is the discrete-time index. It is generally assumed that the signal and noise are uncorrelated. This is a reasonable assumption for most practical cases where the signal and noise are generated by independent sources.

To transform a signal into the frequency domain, the signal samples are divided into overlapping frames with a frame size of N samples. The frame size is limited by the maximum allowable delay of the communication system and by the assumption in the Fourier transform that the signal is stationary. Usually, for audio signals a frame length of about 25 ms is chosen, although in some systems the frame length is varied with the speed of changes in the signal characteristics; shorter length frames are chosen for fast changing signals and longer length frames for more steady signals.

In the frequency domain, the noisy signal equation, Equation (11.1), can be represented as

$$Y(k) = X(k) + N(k) \quad k = 0, \dots, N-1 \quad (11.2)$$

where the complex variables $X(k)$, $N(k)$ and $Y(k)$ are the short time discrete Fourier transforms of speech, noise and noisy speech, respectively, and the integer index k represents the discrete frequency variable; it corresponds to an actual frequency of $2k\pi/N$ rad/s or kF_s/N Hz where F_s is the sampling frequency in Hz.

Equation (11.2), which is in complex Cartesian form, can be rewritten in the complex polar form in terms of the magnitude and the phase of the signal and noise at a discrete frequency k as

$$Y_k e^{j\theta_{Y_k}} = X_k e^{j\theta_{X_k}} + N_k e^{j\theta_{N_k}} \quad k = 0, \dots, N-1 \quad (11.3)$$

where $Y_k = |Y(k)|$ and $\theta_{Y_k} = \tan^{-1} \{ \text{Im} [Y(k)] / \text{Re} [Y(k)] \}$ are the magnitude and phase of the frequency spectrum, respectively. Note that the Fourier transform models the correlation of speech samples with sinusoidal basis functions. The sinusoidal functions can then be processed individually or in groups of frequencies, taking into account the psychoacoustics of hearing.

11.1.2 VECTOR REPRESENTATION OF THE SPECTRUM OF NOISY SIGNALS

The spectrum of a noisy signal, $Y(k)$, is the vector sum of the spectra of the clean signal, $X(k)$, and noise, $N(k)$. The squared spectral amplitude of the noisy signal is given by

$$Y_k^2 = X_k^2 + N_k^2 + 2X_k N_k \cos(\theta) \quad k = 0, \dots, N-1 \quad (11.4)$$

where θ is the angle between the complex spectral vectors of speech, $X(k)$, and noise, $N(k)$. Unless the cross product term, $2X_k N_k \cos(\theta)$, in equation (11.3) is modelled in the spectral amplitude estimation process, it contributes to an estimation error.

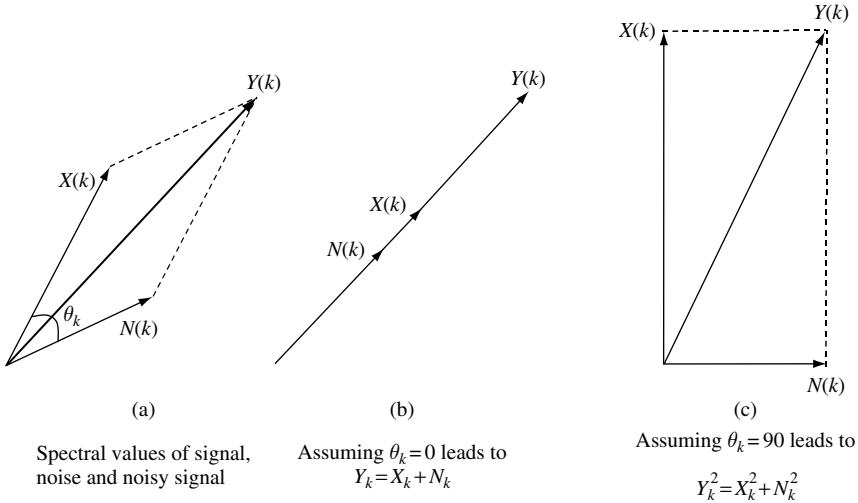


Figure 11.2 (a) Complex spectral vectors of signal, noise and the resultant noisy signal; (b) assuming signal and noise spectral vectors are in phase; (c) assuming signal and noise spectral vectors have a phase difference of $\pi/2$.

Figure 11.2(a) shows the relation between the complex spectral vectors of the signal noisy signal $Y(k)$, the signal $X(k)$ and noise $N(k)$. Figure 11.2(b) shows that the assumption that amplitude spectrum of the noisy signal is the sum of the amplitude spectra of signal and noise is equivalent to assuming that the angle between speech and noise spectral vectors $\theta_k = 0$; this assumption leads to an overestimation of noise or equivalently an underestimation of the signal. Similarly, Figure 11.2(c) shows that the assumption that squared amplitude spectrum (i.e. the instantaneous power spectra) of the noisy signal is the sum of the squared amplitude spectra of signal and noise is equivalent to assuming that the angle between speech and noise spectral vectors is $\theta_k = 90$. This assumption can lead to an underestimation (if $\theta_k < 90$) or overestimation (if $\theta_k > 90$) of the noise.

11.2 SPECTRAL SUBTRACTION

In this spectral subtraction an estimate of the spectral amplitude of the signal is obtained by subtracting an estimate of the spectral amplitude of noise from that of the noisy signal. Figure 11.3 illustrates a block diagram configuration of the spectral subtraction method.

In spectral subtraction, the incoming signal, $x(m)$, is divided into frames of N sample length. Each frame is windowed, using a window (i.e. Hann) and then transformed via discrete Fourier transform to N spectral samples. The windows alleviate the effects of the discontinuities at the endpoints of each segment. The windowed signal is given by

$$\begin{aligned}
 y_w(m) &= w(m)y(m) \\
 &= w(m)[x(m) + n(m)] \\
 &= x_w(m) + n_w(m)
 \end{aligned}
 \tag{11.5}$$

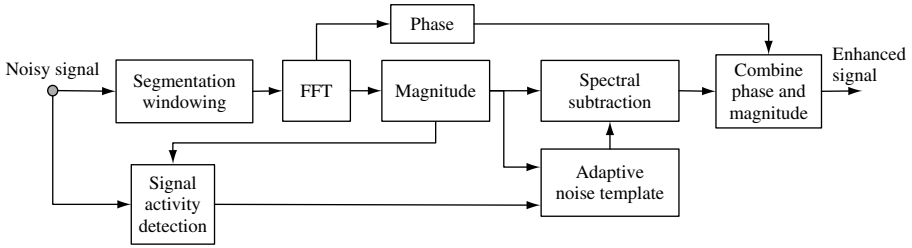


Figure 11.3 Block diagram illustration of an FFT-based spectral subtraction system for de-noising speech.

The windowing operation can be expressed in the frequency domain as

$$\begin{aligned} Y_w(f) &= W(f) * Y(f) \\ &= X_w(f) + N_w(f) \end{aligned} \quad (11.6)$$

where the $*$ denotes convolution operation. Throughout this chapter, it is assumed that the signals are windowed, and hence for simplicity we drop the use of the subscript w for windowed signals.

The equation describing spectral subtraction may be expressed as

$$\hat{X}_k^b = Y_k^b - \alpha(k) \overline{N}_k^b \quad (11.7)$$

where \hat{X}_k^b is an estimate of the signal magnitude spectrum to the power of b and \overline{N}_k^b is the time-averaged magnitude of noise spectra to the power b . It is assumed that the noise is a stationary random process. For magnitude spectral subtraction, the exponent $b = 1$, and for power spectral subtraction, $b = 2$. The parameter $\alpha(k)$ in Equation (11.7) controls the amount of noise subtracted from the noisy signal. For full noise subtraction, $\alpha(k) = 1$ and for over-subtraction $\alpha(k) > 1$. The time-averaged noise spectrum is obtained from the periods when the signal is absent and only the noise is present as

$$\overline{N}_k^b = \frac{1}{M} \sum_{i=0}^{M-1} N_{k,i}^b \quad (11.8)$$

In Equation (11.8), $N_{k,i}$ is the spectrum of the i th noise frame at discrete frequency k , and it is assumed that there are M frames in a noise-only period, where M is a variable. Alternatively, the averaged noise amplitude spectrum can be obtained as the output of a first-order digital low-pass filter as

$$\overline{N}_{k,i}^b = a \overline{N}_{k,i-1}^b + (1-a) N_{k,i}^b \quad (11.9)$$

where the low-pass filter coefficient, a , is typically set between 0.85 and 0.99. For restoration of a time-domain signal, the amplitude spectrum estimate \hat{X}_k is combined with the phase of the noisy signal, and then transformed into the time domain via the inverse discrete Fourier transform as

$$\hat{x}(m) = \sum_{k=0}^{N-1} \left(\hat{X}_k e^{j\theta_{y_k}} \right) e^{j(2\pi/N)km} \quad m = 0, \dots, N-1 \quad (11.10)$$

where θ_{Y_k} is the phase of the noisy signal frequency, $Y(k)$. The signal restoration equation, Equation (11.10), is based on the assumption that the audible noise is mainly due to the distortion of the magnitude spectrum, and that the phase distortion is largely inaudible. Evaluations of the perceptual effects of simulated phase distortions validate this assumption.

Owing to the variations of the noise spectrum, spectral subtraction may result in negative estimates of the power or the amplitude spectrum. This outcome is more probable as the signal-to-noise ratio decreases. To avoid negative magnitude estimates, the spectral subtraction output is post-processed using a mapping function $T[\cdot]$ of the form

$$T[\hat{X}_k] = \begin{cases} \hat{X}_k & \text{if } \hat{X}_k > \beta Y_k \\ fn[Y_k] & \text{otherwise} \end{cases} \quad (11.11)$$

For example, we may choose a rule such that if the estimate $\hat{X}_k > 0.01Y_k$ (in magnitude spectrum 0.01 is equivalent to -40 dB) then \hat{X}_k should be set to some function of the noisy signal $fn[Y_k]$. In its simplest form, $fn[Y_k] = \text{noise floor}$, where the noise floor is a positive constant. An alternative choice is $fn[Y_k] = \beta Y_k$. In this case,

$$T[\hat{X}_k] = \begin{cases} \hat{X}_k & \text{if } \hat{X}_k > \beta Y_k \\ \beta Y_k & \text{otherwise} \end{cases} \quad (11.12)$$

Spectral subtraction may be implemented in the power or the magnitude spectral domains. The two methods are similar, although theoretically they result in somewhat different expected performance.

11.2.1 POWER SPECTRUM SUBTRACTION

The power spectrum subtraction, or squared-magnitude spectrum subtraction, is defined by the following equation:

$$\hat{X}_k^2 = Y_k^2 - \overline{N_k^2} \quad (11.13)$$

where it is assumed that $\alpha(k)$, the subtraction factor in Equation (11.7), is 1. Note that Equation (11.13) can be deduced from Equation (11.4) if we assume that the signal and noise spectral vectors are perpendicular, as illustrated in Figure (11.2.c).

We denote the power spectrum by $\mathcal{E}[X_k^2]$, the time-averaged power spectrum by $\overline{X_k^2}$ and the *instantaneous* power spectrum by X_k^2 . By expanding the instantaneous power spectrum of the noisy signal Y_k^2 , and grouping the appropriate terms, Equation (11.13) may be rewritten as

$$\hat{X}_k^2 = X_k^2 + \underbrace{\left(N_k^2 - \overline{N_k^2}\right)}_{\text{Noise variations}} + \underbrace{X_k^* N_k + X_k N_k^*}_{\text{Cross products}} \quad (11.14)$$

Taking the expectations of both sides of Equation (11.14), and assuming that the signal and the noise processes are uncorrelated ergodic processes, we have

$$\mathcal{E}[\hat{X}_k^2] = \mathcal{E}[X_k^2] \quad (11.15)$$

From Equation (11.15), the average of the estimate of the instantaneous power spectrum converges to the power spectrum of the noise-free signal. However, it must be noted that, for restoration of nonstationary signals, such as speech, the objective is to recover the *instantaneous* or the short-time spectrum, and only a relatively small amount of averaging can be applied. Too much averaging will smear and obscure the temporal evolution of the spectral events. Note that, in deriving Equation (11.15), we have not considered nonlinear rectification of the negative estimates of the squared magnitude spectrum.

11.2.2 MAGNITUDE SPECTRUM SUBTRACTION

The magnitude spectrum subtraction is defined as

$$\hat{X}_k = Y_k - \overline{N}_k \quad (11.16)$$

where \overline{N}_k is the time-averaged magnitude spectrum of the noise. Note that Equation (11.16) can be deduced from Equation (11.4) if we assume that the signal and noise spectral vectors are in phase as illustrated in Figure 11.2(b). Taking the expectation of Equation (11.16), we have

$$\begin{aligned} \hat{X}_k &= \mathcal{E}[Y_k] - \mathcal{E}[\overline{N}_k] \\ &= \mathcal{E}[|X(k) + N(k)|] - \mathcal{E}[|\overline{N}_k|] \\ &\approx \mathcal{E}[X_k] \end{aligned} \quad (11.17)$$

For signal restoration the magnitude estimate is combined with the phase of the noisy signal and then transformed into the time domain using Equation (11.10).

11.2.3 SPECTRAL SUBTRACTION FILTER: RELATION TO WIENER FILTERS

The spectral subtraction equation can be expressed as the product of the noisy signal spectrum and the frequency response of a spectral subtraction filter as

$$\hat{X}_k^2 = Y_k^2 - \overline{N}_k^2 = H_k Y_k^2 \quad (11.18)$$

where H_k , the frequency response of the spectral subtraction filter, at discrete frequency k , is defined as

$$H_k = 1 - \frac{\overline{N}_k^2}{Y_k^2} = \frac{Y_k^2 - \overline{N}_k^2}{Y_k^2} \quad (11.19)$$

The spectral subtraction filter, H_k , is a zero-phase filter, with its magnitude response in the range $0 \leq H_k \leq 1$. The filter acts as an SNR-dependent attenuator. The attenuation at each frequency increases with the decreasing SNR, and conversely decreases with the increasing SNR.

The least mean square error linear filter for noise removal is the Wiener filter, covered in Chapter 6. Implementation of a Wiener filter requires the power spectra (or equivalently the correlation functions) of the signal and the noise process, as discussed in Chapter 6. Spectral subtraction is used as a substitute for the Wiener filter when the signal power spectrum is not available. In this section, we discuss the close relation between the Wiener filter and spectral subtraction. For restoration of a signal observed in uncorrelated additive noise, the equation describing the frequency response of the Wiener filter was derived in Chapter 6 as

$$W_k = \frac{\mathcal{E}[Y_k^2] - \mathcal{E}[N_k^2]}{\mathcal{E}[Y_k^2]} \quad (11.20)$$

A comparison of W_k and H_k , from Equations (11.20) and (11.19), shows that the Wiener filter is based on the *ensemble-average* spectra of the signal and the noise, whereas the spectral subtraction filter uses the instantaneous spectra of the noisy signal and the *time-averaged* spectra of the noise. In spectral subtraction, we only have access to a single realisation of the signal process. However, assuming that the signal and noise are wide-sense stationary ergodic processes, we may replace the instantaneous noisy signal spectrum, Y_k^2 , in the spectral subtraction equation [Equation (11.20)] with the time-averaged spectrum, $\overline{Y_k^2}$, to obtain

$$H_k = \frac{\overline{Y_k^2} - \overline{N_k^2}}{\overline{Y_k^2}} \quad (11.21)$$

For an ergodic process, as the length of the time over which the signals are averaged increases, the time-averaged spectrum approaches the ensemble-averaged spectrum and, in the limit, the spectral subtraction filter of Equation (11.21) approaches the Wiener filter equation, Equation (11.20). In practice, many signals, such as speech and music, are nonstationary, and only a limited degree of beneficial time-averaging of the spectral parameters can be expected.

11.2.4 PROCESSING DISTORTIONS

The main problem in spectral subtraction is the nonlinear processing distortions caused by the random variations of the noise spectrum. From Equation (11.12) and the constraint that the magnitude spectrum must have a non-negative value, we may identify three sources of distortions of the instantaneous estimate of the magnitude or power spectrum as:

- (1) the variations of the instantaneous noise power spectrum about the mean;
- (2) the signal and noise cross-product terms;
- (3) the nonlinear mapping of the spectral estimates that fall below a threshold.

The same sources of distortion appear in both the magnitude and the power spectrum subtraction methods. Of the three sources of distortion listed above, the dominant distortion is often due to the nonlinear mapping of the negative, or small-valued, spectral estimates. This distortion produces a metallic sounding noise, known as ‘musical tone noise’ due to its narrow-band spectrum and the tin-like sound. The success of spectral subtraction depends on the ability of the algorithm to reduce the noise variations and to remove the processing

distortions. In its worst, and not uncommon, case the residual noise can have the following two forms:

- (1) a sharp trough or peak in the signal spectra;
- (2) isolated narrow bands of frequencies.

In the vicinity of a high amplitude signal frequency, the noise-induced trough or peak is often masked, and made inaudible, by the high signal energy. The main cause of audible degradations is the isolated frequency components also known as ‘musical tones’ or ‘musical noise’, illustrated in Figure 11.4. The musical noise is characterised as short-lived narrow bands of frequencies surrounded by relatively low-level frequency components. In audio signal restoration, the distortion caused by spectral subtraction can result in a significant deterioration of the signal quality. This is particularly true at low signal-to-noise ratios. The effects of a bad implementation of subtraction algorithm can result in a signal that is of a lower perceived quality, and lower information content, than the original noisy signal.

11.2.5 EFFECT OF SPECTRAL SUBTRACTION ON SIGNAL DISTRIBUTION

Figure 11.5 is an illustration of the distorting effect of spectral subtraction on the distribution of the magnitude spectrum of a signal. In this figure, we have considered the simple case where the spectrum of a signal is divided into two parts: a low-frequency band, f_l , and a high-frequency band, f_h . Each point in Figure 11.5 is a plot of the high-frequency spectrum vs the low-frequency spectrum, in a two-dimensional signal space. Figure 11.5(a) shows an assumed distribution of the spectral samples of a signal in the two-dimensional magnitude–frequency space. The effect of the random noise, shown in Figure 11.5(b), is an increase in the mean and the variance of the spectrum, by an amount that depends on the mean and the variance of the magnitude spectrum of the noise. The increase in the variance constitutes an irrevocable distortion. The increase in the mean of the magnitude spectrum can be removed through spectral subtraction. Figure 11.5(c) illustrates the distorting effect

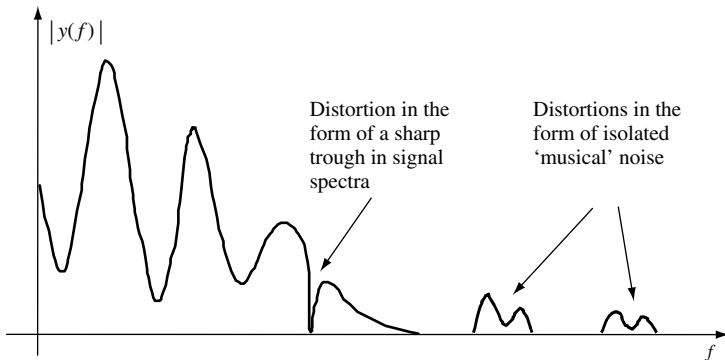


Figure 11.4 Illustration of distortions that may result from spectral subtraction.

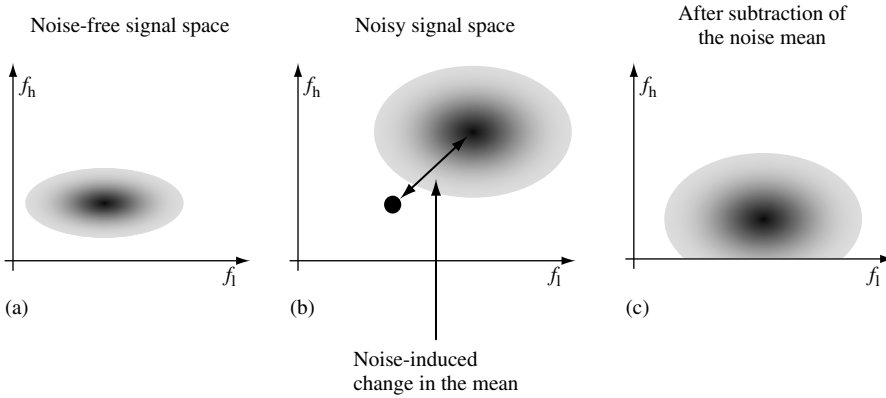


Figure 11.5 Illustration of the distorting effect of spectral subtraction on the space of the magnitude spectrum of a signal.

of spectral subtraction on the distribution of the signal spectrum. As shown, owing to the noise-induced increase in the variance of the signal spectrum, after subtraction of the average noise spectrum, a proportion of the signal population, particularly that with a low SNR, becomes negative and has to be mapped to non-negative values. As shown, this process distorts the distribution of the low-SNR part of the signal spectrum.

11.2.6 REDUCING THE NOISE VARIANCE

The distortions that result from spectral subtraction are due to the variations in the noise spectrum. In Section 9.4 we considered the mean and variance of the estimate of a power spectrum. For a white noise process with variance σ_n^2 , it can be shown that the variance of the DFT spectrum of the noise N_k is given by

$$\text{Var} [N_k^2] \approx P_{NN}^2(k) = \sigma_n^4 \tag{11.22}$$

and the variance of the running average of K independent spectral components is

$$\text{Var} \left[\frac{1}{K} \sum_{i=0}^{K-1} N_{k,i}^2 \right] \approx \frac{1}{K} P_{NN}^2(k) \approx \frac{1}{K} \sigma_n^4 \tag{11.23}$$

From Equation (11.23), the noise variations can be reduced by time-averaging of the noisy signal frequency components. The fundamental limitation is that the averaging process, in addition to reducing the noise variance, also has the undesirable effect of smearing and blurring the time variations of the signal spectrum. Therefore an averaging process should reflect a compromise between the conflicting requirements of reducing the noise variance and of retaining the time resolution of the nonstationary spectral events. This is important because time resolution plays an important part in both the quality and the intelligibility of audio signals.

In spectral subtraction, the noisy signal, $y(m)$, is segmented into frames of N samples. Each signal frame is then transformed via a DFT into N spectral samples, $Y(k)$. Successive frames of spectral samples form a two-dimensional frequency–time matrix denoted by $Y(k, i)$ where the integer variable, i , is the frame index and denotes the time dimension. The signal $Y(k, i)$ can be considered as a band-pass channel, k , that contains a time-varying signal $X(k, i)$ plus a random noise component $N(k, i)$. One method for reducing the noise variations is to low-pass filter the magnitude spectrum at each frequency. A simple recursive first-order digital low-pass filter is given by

$$\overline{Y}_{k,i} = \rho \overline{Y}_{k,i-1} + (1 - \rho) Y_{k,i} \tag{11.24}$$

where the $\overline{Y}_{k,i}$ is the output of the low-pass filter, and the smoothing coefficient ρ controls the bandwidth and the time constant of the low-pass filter.

11.2.7 FILTERING OUT THE PROCESSING DISTORTIONS

Audio signals, such as speech and music, are composed of sequences of nonstationary acoustic events. The acoustic events are ‘born’, have a varying lifetime, disappear, and then reappear with a different intensities and spectral composition. The time- varying nature of audio signals plays an important role in conveying information, sensation and quality. The musical tone noise, introduced as an undesirable by-product of spectral subtraction, is also time-varying. However, there are significant differences between the characteristics of most audio signals and so-called ‘musical noise’. The characteristic differences may be used to identify and remove some of the more annoying distortions. Identification of musical noise may be achieved by examining the variations of the signal in the time and frequency domains. The main characteristics of musical noise are that it tends to be relatively short-lived random isolated bursts of narrow band signals, with relatively small amplitudes.

Using a DFT block size of 128 samples, at a sampling rate of 20 kHz, experiments indicate that the great majority of musical noise tends to last no more than three frames, whereas genuine signal frequencies have a considerably longer duration. This observation was used as the basis of an effective ‘musical noise’ suppression system. Figure 11.6 demonstrates a

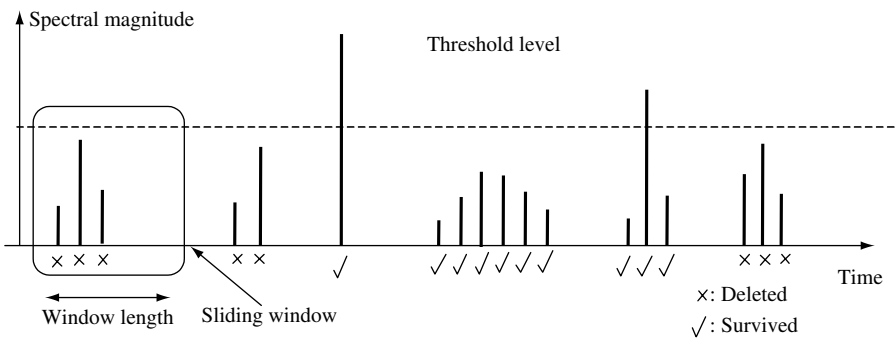


Figure 11.6 Illustration of a method for identification and filtering of ‘musical noise’.

method for the identification of musical noise. Each DFT channel is examined to identify short-lived frequency events. If a frequency component has a duration shorter than a pre-selected time window, and an amplitude smaller than a threshold, and is not masked by signal components in the adjacent frequency bins, then it is classified as distortion and deleted.

11.2.8 NONLINEAR SPECTRAL SUBTRACTION

The use of spectral subtraction in its basic form of Equation (11.7) may cause deterioration in the quality and the information content of a signal. For example, in audio signal restoration, the musical noise can cause degradation in the perceived quality of the signal, and in speech recognition the basic spectral subtraction can result in deterioration of the recognition accuracy. In the literature, there are many variants of spectral subtraction that aim to provide consistent performance improvement across a range of SNRs. These methods differ in their approach to estimation of the noise spectrum, in their method of averaging the noisy signal spectrum, and in their post-processing method for the removal of processing distortions.

Nonlinear spectral subtraction methods are heuristic methods that utilise estimates of the local SNR and the observation that, at a low SNR, oversubtraction can produce improved results. For an explanation of the improvement that can result from oversubtraction, consider the following expression of the basic spectral subtraction equation:

$$\begin{aligned}\hat{X}_k &= Y_k - \overline{N}_k \\ &\approx X_k + N_k - \overline{N}_k \\ &\approx X_k + V_N(k)\end{aligned}\tag{11.25}$$

where $V_N(k)$ is the zero-mean random component of the noise spectrum. If $V_N(k)$ is well above the signal, X_k , then the signal may be considered as lost to noise. In this case, oversubtraction, followed by nonlinear processing of the negative estimates, results in a higher overall attenuation of the noise. This argument explains why subtracting more than the noise average can sometimes produce better results. The nonlinear variants of spectral subtraction may be described by the following equation:

$$\hat{X}_k = Y_k - \alpha[\text{SNR}(k)]\overline{N}_k\tag{11.26}$$

where $\alpha[\text{SNR}(k)]$ is an SNR-dependent subtraction factor and \overline{N}_k is an estimate of the spectral amplitude of noise. The spectral amplitude estimate is further processed to avoid negative estimates as

$$\hat{X}_k = \begin{cases} \hat{X}_k & \text{if } \hat{X}_k > \beta Y_k \\ \beta Y_k & \text{otherwise} \end{cases}\tag{11.27}$$

One form of an SNR-dependent subtraction factor for Equation (11.26) is given by

$$\alpha[\text{SNR}(k)] = 1 + \frac{\text{sd}(N_k)}{\overline{N}_k} \quad (11.28)$$

where the function $\text{sd}(N_k)$ is the standard deviation of the noise at discrete frequency k . For white noise, $\text{sd}(N_k) = \sigma_n$, where σ_n^2 is the noise variance. Substitution of Equation (11.28) in Equation (11.26) yields

$$\hat{X}_k = Y_k - \left[1 + \frac{\text{sd}(N_k)}{\overline{N}_k} \right] \overline{N}_k \quad (11.29)$$

In Equation (11.29) the subtraction factor depends on the mean and the variance of the noise. Note that the amount oversubtracted is the standard deviation of the noise. This heuristic formula is appealing because, at one extreme for deterministic noise with a zero variance, such as a sine wave, $\alpha[\text{SNR}(f)] = 1$, and, at the other extreme for white noise, $\alpha[\text{SNR}(f)] = 2$. In application of spectral subtraction to speech recognition, it is found that the best subtraction factor is usually between 1 and 2.

In the nonlinear spectral subtraction method of Lockwood and Boudy (1992), the spectral subtraction filter is obtained from

$$H_k = \frac{\overline{Y}_k^2 - \text{NL}(\overline{N}_k^2)}{\overline{Y}_k^2} \quad (11.30)$$

Lockwood and Boudy suggested the following function as a nonlinear estimator of the noise spectrum:

$$\text{NL}(\overline{N}_k^2) = \Phi \left[\max_{\text{over } M \text{ frames}} (N_k^2), \text{SNR}(k), \overline{N}_k^2 \right] \quad (11.31)$$

The estimate of the noise spectrum is a function of the maximum value of noise spectrum over M frames, and the signal-to-noise ratio. One form for the nonlinear function $\Phi(\cdot)$ is given by the following equation:

$$\Phi \left[\max_{\text{over } M \text{ frames}} (N_k^2), \text{SNR}(k) \right] = \frac{\max_{\text{over } M \text{ frames}} (N_k^2)}{1 + \gamma \text{SNR}(k)} \quad (11.32)$$

where γ is a design parameter. From Equation (11.32), as the SNR decreases, the output of the nonlinear estimator $\Phi(\cdot)$ approaches $\max(N_k^2)$ and, as the SNR increases, it approaches zero. For oversubtraction, the noise estimate is forced to be an overestimation by using the following limiting function:

$$\overline{N}_k^2 \leq \Phi \left[\max_{\text{over } M \text{ frames}} (N_k^2), \text{SNR}(k), \overline{N}_k^2 \right] \leq 3\overline{N}_k^2 \quad (11.33)$$

The maximum attenuation of the spectral subtraction filter is limited to $H_k \geq \beta$, where usually the lower bound $\beta \geq 0.01$. Figure 11.7 illustrates the effects of nonlinear spectral subtraction and smoothing in restoration of the spectrum of a speech signal.

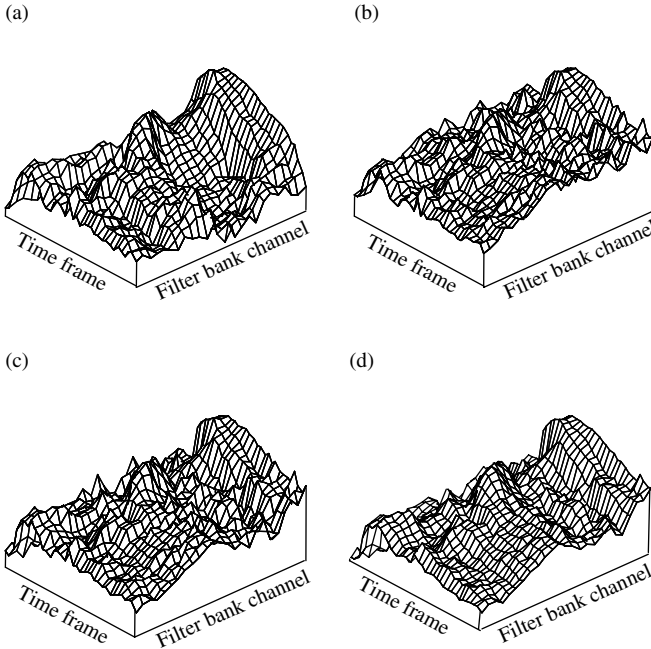


Figure 11.7 Illustration of the effects of nonlinear spectral subtraction. (a) Original clean speech; (b) noisy speech at 12 dB; (c) nonlinear spectral subtraction; (d) nonlinear spectral subtraction with smoothing.

11.2.9 IMPLEMENTATION OF SPECTRAL SUBTRACTION

Figure 11.8 is a block diagram illustration of a spectral subtraction system. It includes the following subsystems:

- (1) a silence detector for detection of the periods of signal inactivity – the noise spectra is updated during these periods;
- (2) a discrete Fourier transformer for transforming the time domain signal to the frequency domain – the DFT is followed by a magnitude operator;
- (3) a lowpass filter for reducing the noise variance – the purpose of the LPF is to reduce the processing distortions due to noise variations;
- (4) a post-processor for removing the processing distortions introduced by spectral subtraction;
- (5) an inverse discrete Fourier transform (IDFT) for transforming the processed signal to the time domain;
- (6) an attenuator, γ , for attenuation of the noise during silent periods.

The DFT-based spectral subtraction is a block processing algorithm. The incoming audio signal is buffered and divided into overlapping blocks of N samples, as shown in Figure 11.8. Each block is windowed, and then transformed via a DFT to the frequency domain. After

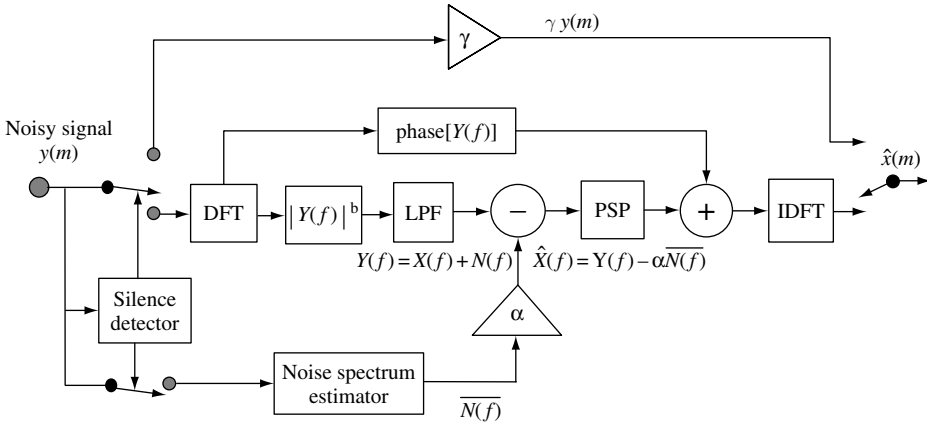


Figure 11.8 Block diagram configuration of a spectral subtraction system. PSP = post-spectral-subtraction processing.

spectral subtraction, the magnitude spectrum is combined with the phase of the noisy signal, and transformed back to the time domain. Each signal block is then overlapped and added to the preceding and succeeding blocks to form the final output.

The choice of the block length for spectral analysis is a compromise between the conflicting requirements of the time resolution and the spectral resolution. Typically a block length of 5–50 ms is used. At a sampling rate of, say, 20 kHz, this translates to a value for N in the range 100–1000 samples. The frequency resolution of the spectrum is directly proportional to the number of samples, N . A larger value of N produces a better estimate of the spectrum. This is particularly true for the lower part of the frequency spectrum, since low-frequency components vary slowly with time, and require a larger window for a stable estimate. The conflicting requirement is that, owing to the nonstationary nature of audio signals, the window length should not be too large, so that short-duration events are not obscured.

The main function of the window and the overlap operations (Figure 11.9) is to alleviate discontinuities at the endpoints of each output block. Although there are a number of useful windows with different frequency/time characteristics, in most implementations of

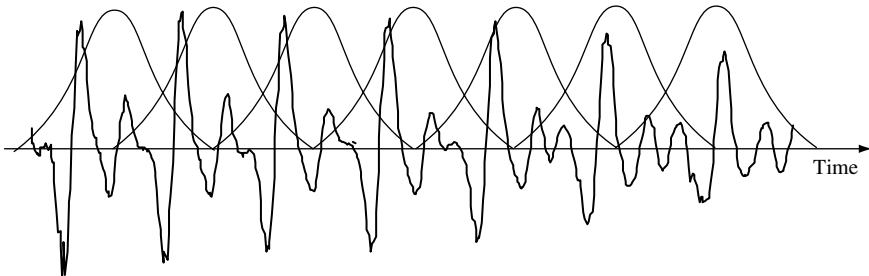


Figure 11.9 Illustration of the window and overlap process in spectral subtraction.

the spectral subtraction, a Hanning window is used. In removing distortions introduced by spectral subtraction, the post-processor algorithm makes use of such information as the correlation of each frequency channel from one block to the next, and the durations of the signal events and the distortions. The correlation of the signal spectral components, along the time dimension, can be partially controlled by the choice of the window length and the overlap. The correlation of spectral components along the time domain increases with decreasing window length and increasing overlap. However, increasing the overlap can also increase the correlation of noise frequencies along the time dimension.

11.3 BAYESIAN MMSE SPECTRAL AMPLITUDE ESTIMATION

Bayesian estimation, covered in Chapter 4, employs the probability density functions of the signal and noise and minimises a cost of error function. The probabilistic minimum mean squared error estimation of the short-time spectral amplitude (STSA) is an example of Bayesian estimation method with a mean squared error cost function.

The noisy signal model in time and frequency is expressed in Equations (11.2–11.4). Assume each frame of N samples of noisy speech signal $y(m)$ is converted into N complex spectral samples, $Y(k) = Y_k e^{j\theta_{Yk}}$ via DFT. The MMSE estimation of the short-time spectral amplitude of clean speech, \hat{X}_k , is derived from the minimization of a mean squared error cost function integral as

$$\hat{X}_k = \underset{\hat{X}_k}{\text{Min}} \int_{-\infty}^{\infty} |X_k - \hat{X}_k|^2 p[X(k)|Y(k)] dX(k) \quad (11.34)$$

where $X_k - \hat{X}_k$ is the spectral amplitude estimation error and $p[X(k)|Y(k)]$ is the posterior probability density function of the spectrum of clean speech $X(k)$ given the spectrum of noisy observation $Y(k)$.

Setting the derivative w.r.t. \hat{X}_k of the integral of the mean squared error cost function in Equation (11.34) to zero (see Chapter 4), we obtain the MMSE STSA estimate as

$$\hat{X}_k = E[X_k|Y(k)] = \int_{-\infty}^{\infty} X_k p[X(k)|Y(k)] dX(k) \quad k = 0, 1, \dots, N-1 \quad (11.35)$$

where $E[\cdot]$ is the expectation or averaging operator. *Note that the MMSE estimate of a variable is the mean or expectation of the variable.*

Using Bayes's rule, $p[X(k)|Y(k)] = p[Y(k)|X(k)]p[X(k)]/p[Y(k)]$ in Equation (11.35), we obtain

$$\hat{X}_k = \frac{\int_{-\infty}^{\infty} X_k p[Y(k)|X(k)] p[X(k)] dX(k)}{\int_{-\infty}^{\infty} p[Y(k)|X(k)] dX(k)} \quad (11.36)$$

Now in Equation (11.36) the probability of each spectral component $X(k)$ of the clean signal can be written in terms of the probability of its magnitude, X_k , and phase, θ_{X_k} , $p[X(k)] = p(X_k, \theta_{X_k})$, as

$$\hat{X}_k = \frac{\int_{-\infty}^{\infty} \int_0^{2\pi} X_k p[Y(k)|X_k, \theta_{X_k}] p(X_k, \theta_{X_k}) d\theta_{X_k} dX_k}{\int_{-\infty}^{\infty} \int_0^{2\pi} p[Y(k)|X_k, \theta_{X_k}] d\theta_{X_k} dX_k} \quad (11.37)$$

The MMSE Equation (11.37) requires the likelihood of the noisy speech $p[Y(k)|X_k, \theta_{X_k}]$ and the prior probability density function of the clean speech, $p(X_k, \theta_{X_k})$.

Assuming that the magnitude and phase of speech spectrum $X(k)$ are independent, and that the phase of the speech spectrum has a uniform distribution with a probability of $p(\theta_{X_k}) = \frac{1}{2\pi}$, we have

$$p(X_k, \theta_{X_k}) = p(\theta_{X_k}) p(X_k) = \frac{1}{2\pi} p(X_k) \quad (11.38)$$

Assuming that the complex spectrum of the clean signal, $X(k)$, has a Gaussian distribution in the Cartesian complex domain defined as

$$p[X(k)] = \frac{1}{\pi\sigma^2} \exp\left(-\frac{\{\text{Re}[X(k)]\}^2 + \{\text{Im}[X(k)]\}^2}{2\sigma^2}\right) \quad (11.39)$$

where Re and Im denote the real and imaginary parts of a complex variable, then it can be shown (see Chapter 4) that X_k , the magnitude of $X(k)$, has a Rayleigh distribution defined as

$$p(X_k) = \frac{X_k}{\sigma^2} \exp\left(-\frac{X_k^2}{2\sigma^2}\right) \quad (11.40)$$

where σ^2 is the variance of X_k . As already explained, the phase of $X(k)$ is assumed to have a uniform distribution.

Ephraim and Malah derived an MMSE spectral amplitude estimation algorithm using a Rayleigh distribution for the magnitude spectrum of clean speech, a uniform distribution for the phase of the clean speech and a complex Gaussian distribution for noisy speech. The resulting estimator is of the form

$$\hat{X}_k = \Gamma(1.5) \frac{\sqrt{v_k}}{\gamma_k} \exp\left(-\frac{v_k}{2}\right) \left[(1+v_k) I_0\left(-\frac{v_k}{2}\right) + v_k I_1\left(-\frac{v_k}{2}\right) \right] Y_k \quad (11.41)$$

where $\Gamma(\cdot)$ is the gamma function, $I_n(\cdot)$ is Bessel function of order n and v_k and γ_k are defined as

$$v_k = \frac{\xi(k)}{1 + \xi(k)} \gamma_k \quad (11.42)$$

$$\xi_k = \frac{\sigma_X^2(k)}{\sigma_N^2(k)} \quad (11.43)$$

$$\gamma_k = \frac{Y^2(k)}{\sigma_N^2(k)} \quad (11.44)$$

where $\sigma_x^2(k)$ and $\sigma_N^2(k)$ are the variance of speech and noise spectra, ξ_k is known as the prior signal-to-noise ratio and γ_k is known as the posterior signal to noise ratio. For high SNR, Equation (11.44) tends to the Wiener solution.

The minimum mean squared estimate of spectral amplitude can be expressed as

$$\hat{X}_k = G_{\text{MMSE}}(k) Y_k \quad (11.45)$$

where the spectral gain factor $G_{\text{MMSE}}(k)$ is given by

$$G_{\text{MMSE}}(k) = \Gamma(1.5) \frac{\sqrt{v_k}}{\gamma_k} \exp\left(-\frac{v_k}{2}\right) \left[(1+v_k) I_0\left(-\frac{v_k}{2}\right) + v_k I_1\left(-\frac{v_k}{2}\right) \right] \quad (11.46)$$

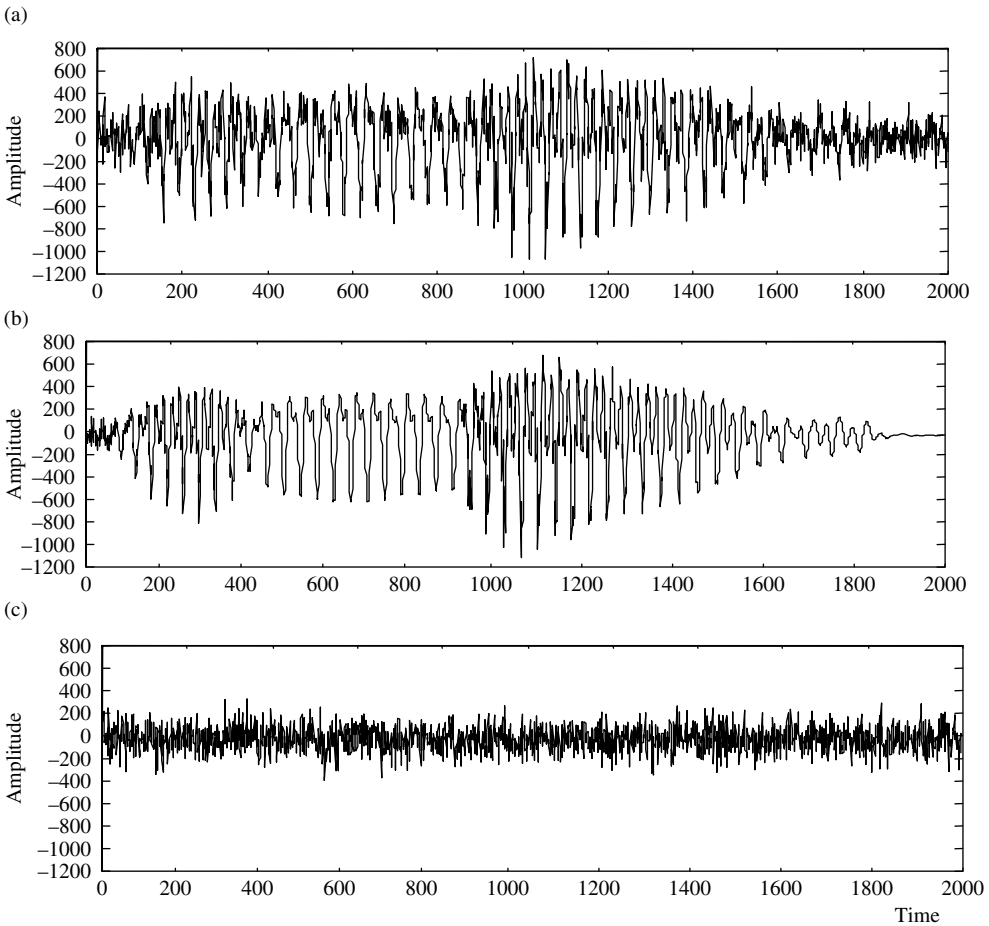


Figure 11.10 (a) A noisy signal; (b) restored signal after spectral subtraction; (c) noise estimate obtained by subtracting (b) from (a).

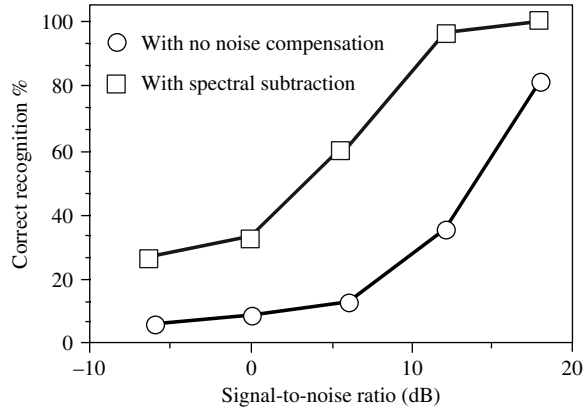


Figure 11.11 The effect of spectral subtraction in improving speech recognition (for a spoken digit data base) in the presence of helicopter noise.

11.4 APPLICATION TO SPEECH RESTORATION AND RECOGNITION

In speech restoration, the objective is to estimate the instantaneous signal spectrum, $X(f)$. The restored magnitude spectrum is combined with the phase of the noisy signal to form the restored speech signal. In contrast, speech recognition systems are more concerned with the restoration of the envelope of the short-time spectrum than the detailed structure of the spectrum. Averaged values, such as the envelope of a spectrum, can often be estimated with more accuracy than the instantaneous values. However, in speech recognition, as in signal restoration, the processing distortion due to the negative spectral estimates can cause substantial deterioration in performance. A careful implementation of spectral subtraction can result in a significant improvement in the recognition performance.

Figure 11.10 illustrates the effects of spectral subtraction in restoring a section of a speech signal contaminated with white noise. Figure 11.11 illustrates the improvement that can be obtained from application of spectral subtraction to recognition of noisy speech contaminated by helicopter noise. The recognition results were obtained for a hidden Markov model-based spoken digit recognition.

11.5 SUMMARY

This chapter began with an introduction to spectral subtraction and its relation to Wiener filters. The main attraction of spectral subtraction is its relative simplicity, in that it only requires an estimate of the noise power spectrum. However, this can also be viewed as a fundamental limitation in that spectral subtraction does not utilise the statistics and the distributions of the signal process. The main problem in spectral subtraction is the presence of processing distortions caused by the random variations of the noise. The estimates of the magnitude and power spectral variables that, owing to noise variations, are negative have to be mapped into non-negative values. In Section 11.2 we considered the processing distortions,

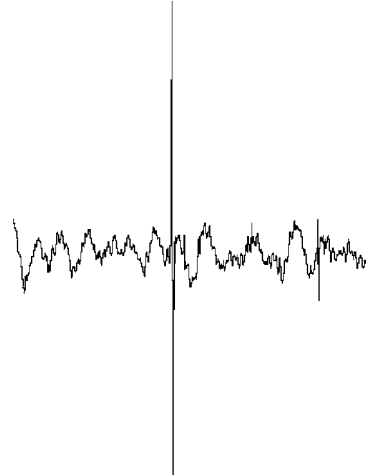
and illustrated the effects of rectification of negative estimates on the distribution of the signal spectrum. In Section 11.3 a number of nonlinear variants of the spectral subtraction method were considered. In signal restoration and in applications of spectral subtraction to speech recognition, it is found that oversubtraction, which is subtracting more than the average noise value, can lead to improved results; if a frequency component is immersed in noise, then oversubtraction can cause further attenuation of the noise. A formula is proposed in which the oversubtraction factor is made dependent on the noise variance. As mentioned earlier, the fundamental problem with spectral subtraction is that it employs relatively too little prior information, and for this reason it is outperformed by Wiener filters and Bayesian statistical restoration methods.

BIBLIOGRAPHY

- Boll S.F. (1979) Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-27**(2): 113–120.
- Brouti M., Schwartz R. and Makhoul J. (1979) Enhancement of speech corrupted by acoustic noise. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing, ICASSP-79*, pp. 208–211.
- Cappe O. (1994) Elimination of musical noise phenomenon with the Ephraim and Malah noise suppressor. *IEEE Trans. Speech Audio Process.* **2**(2): 345–349.
- Crozier P.M., Crozier B., Cheethan M.G. and Holt C. (1993) The use of linear prediction and spectral scaling for improving speech enhancement. *EuroSpeech-93*, pp. 231–234.
- Ephraim Y. (1992) Statistical model-based speech enhancement systems. *Proc. IEEE*, **80**(10): 1526–1555.
- Ephraim Y. and Malah D. (1984) Speech enhancement using a minimum mean-square error short-time amplitude estimator. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP- 32**(6): 1109–1121.
- Ephraim Y. and Van Trees H.L. (1993) A signal subspace approach for speech enhancement. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing, ICASSP-93*, pp. 355–358.
- Juang B.H. and Rabiner L.R. (1987) Signal restoration by spectral mapping. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing, ICASSP- 87*, Texas, pp. 2368–2371.
- Kobayashi T. *et al* (1993) Speech recognition under the non-stationary noise based on the noise hidden Markov model and spectral subtraction. *EuroSpeech-93*, pp. 833–837.
- Lim J.S. (1978) Evaluations of correlation subtraction method for enhancing speech degraded by additive white noise. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP- 26**(5): 471–472.
- Linhard K. and Klemm H. (1997) Noise reduction with spectral subtraction and median filtering for suppression of musical tones. *Proceedings of ECSA-NATO Workshop on Robust Speech Recognition*, pp. 159–162.
- Lockwood P. and Boudy J. (1992) Experiments with a non-linear spectral subtractor (NSS) hidden Markov models and the projection, for robust speech recognition in car. In *Speech Communications*. Elsevier, Amsterdam, pp. 215–228.
- Lockwood P. *et al* (1992) Non-linear spectral subtraction and hidden Markov models for robust speech recognition in car noise environments. *ICASSP-92*, pp. 265–268.
- McAulay R.J. and Malpass M.L. (1980) Speech enhancement using a soft-decision noise suppression filter. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-28**(2): 137–145.
- Milner B.P. (1995) Speech recognition in adverse environments. Ph.D. Thesis, University of East Anglia, Norwich.
- Nolazco-Flores J. and Young S. (1993) Adapting a HMM-based recogniser for noisy speech enhanced by spectral subtraction. *Proceedings of Eurospeech*, pp. 829–832.
- Porter J.E. and Boll S.F. (1984) Optimal estimators for spectral restoration of noisy speech. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing, ICASSP-84*, pp. 18A.2.1–18A.2.4.
- O’Shaughnessy D. (1989) Enhancing speech degraded by additive noise or interfering speakers. *IEEE Commun. Mag.*, 46–52.
- Pollak P. *et al* (1993) Noise suppression system for a car. *EuroSpeech-93*, pp. 1073–1076.
- Sondhi M.M., Schmidt C.E. and Rabiner R. (1981) Improving the quality of a noisy speech signal. *Bell Syst. Tech. J.*, **60**(8): 1847–1859.
- Sorenson H.B. (1993) Robust speaker independent speech recognition using non-linear spectral subtraction based IMELDA. *EuroSpeech-93*, pp. 235–238.

- Van Compernelle D. (1989) Noise adaptation in a hidden Markov model speech recognition system. *Comput. Speech Lang.*, **3**: 151–167.
- Vaseghi S.V. and Frayling-Corck R. (1993) Restoration of archived gramophone records. *J. Audio Enging Soc.* **40**(10): 791–801.
- Xie F. (1993) Speech enhancement by non-linear spectral estimation: a unifying approach. *EuroSpeech-93*, pp. 617– 620.
- Zwicker E. and Fastel H. (1999) *Psychoacoustics, Facts and Models*, 2nd edn. Springer, Berlin.

12



Impulsive Noise

12.1 Impulsive Noise	320
12.2 Statistical Models for Impulsive Noise	323
12.3 Median Filters	327
12.4 Impulsive Noise Removal Using Linear Prediction Models	328
12.5 Robust Parameter Estimation	333
12.6 Restoration of Archived Gramophone Records	334
12.7 Summary	335
Bibliography	336

Impulsive noise consists of relatively short duration ‘on/off’ noise pulses, caused by a variety of sources, such as switching noise, adverse channel environments in a communication system, dropouts or surface degradation of audio recordings, clicks from computer keyboards, etc. An impulsive noise filter can be used for enhancing the quality and intelligibility of noisy signals, and for achieving robustness in pattern recognition and adaptive control systems. This chapter begins with a study of the frequency–time characteristics of impulsive noise, and then proceeds to consider several methods for statistical modelling of an impulsive noise process. The classical method for removal of impulsive noise is the median filter. However, the median filter often results in some signal degradation. For optimal performance, an impulsive noise removal system should utilise (a) the distinct features of the noise and the

signal in the time and/or frequency domains, (b) the statistics of the signal and the noise processes and (c) a model of the physiology of the signal and noise generation. We describe a model-based system that detects each impulsive noise and then proceeds to replace the samples obliterated by an impulse. We also consider some methods for introducing robustness to impulsive noise in parameter estimation.

12.1 IMPULSIVE NOISE

In this section, first the mathematical concepts of an analogue and a digital impulse are introduced, and then the various forms of real impulsive noise in communication systems are considered.

The mathematical concept of an analogue impulse is illustrated in Figure 12.1. Consider the unit-area pulse, $p(t)$, shown in Figure 12.1(a). As the pulse width, Δ , tends to zero, the pulse tends to an impulse. The impulse function shown in Figure 12.1(b) is defined as a pulse with an infinitesimal time width as

$$\delta(t) = \lim_{\Delta \rightarrow 0} p(t) = \begin{cases} 1/\Delta, & |t| \leq \Delta/2 \\ 0, & |t| > \Delta/2 \end{cases} \quad (12.1)$$

The integral of the impulse function is given by

$$\int_{-\infty}^{\infty} \delta(t) dt = \Delta \times \frac{1}{\Delta} = 1 \quad (12.2)$$

The Fourier transform of the impulse function is obtained as

$$\Delta(f) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft} dt = e^0 = 1 \quad (12.3)$$

where f is the frequency variable. The impulse function is used as a *test function* to obtain the impulse response of a system. This is because, as shown in Figure 12.1(c), *an impulse is a spectrally rich signal containing all frequencies in equal amounts*.

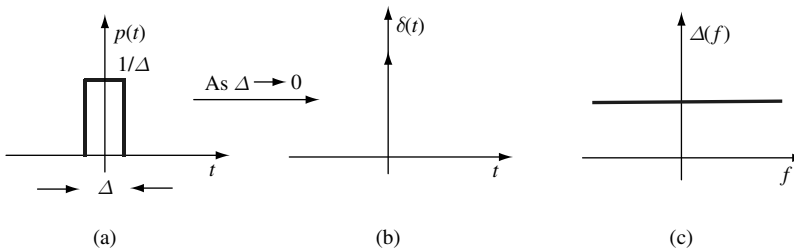


Figure 12.1 (a) A unit-area pulse; (b) the pulse becomes an impulse as $\Delta \rightarrow 0$; (c) the spectrum of the impulse function.

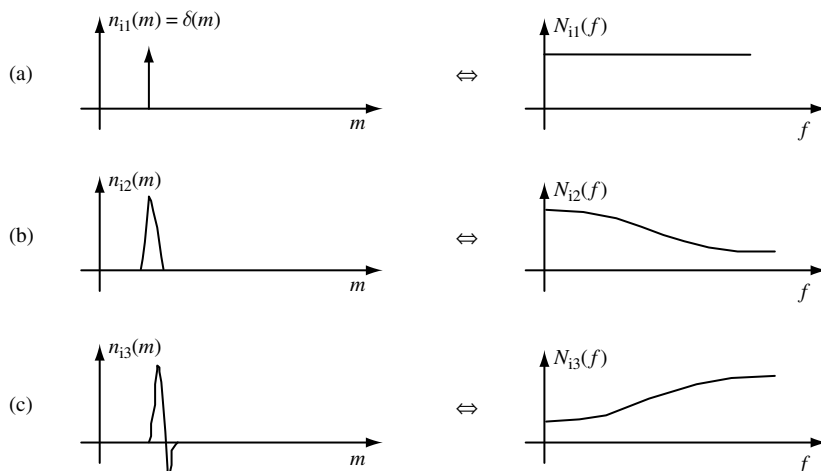


Figure 12.2 Time and frequency sketches of (a) an ideal impulse, and (b) and (c) short-duration pulses.

A digital impulse, $\delta(m)$, shown Figure 12.2(a), is defined as a signal with an ‘on’ duration of one sample, and is expressed as

$$\delta(m) = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0 \end{cases} \tag{12.4}$$

where the variable m designates the discrete-time index. Using the Fourier transform relation, the frequency spectrum of a digital impulse is given by

$$\Delta(f) = \sum_{m=-\infty}^{\infty} \delta(m)e^{-j2\pi fm} = 1.0 - \infty < f < \infty \tag{12.5}$$

In communication systems, real impulsive-type noise has a duration that is normally more than one sample long. For example, in the context of audio signals, short-duration, sharp pulses, of up to 3 ms (60 samples at a 20 kHz sampling rate) may be considered as impulsive-type noise. Figure 12.2(b) and (c) illustrates two examples of short-duration pulses and their respective spectra.

In a communication system, an impulsive noise originates at some point in time and space, and then propagates through the channel to the receiver. The received noise is shaped by the channel, and can be considered as the channel impulse response. In general, the characteristics of a communication channel may be linear or nonlinear, stationary or time-varying. Furthermore, many communication systems, in response to a large-amplitude impulse, exhibit a nonlinear character.

Figure 12.3 illustrates some examples of impulsive noise, typical of those observed on an old gramophone recording. In this case, the communication channel is the playback system, and may be assumed to be time-invariant. The figure also shows some variations of

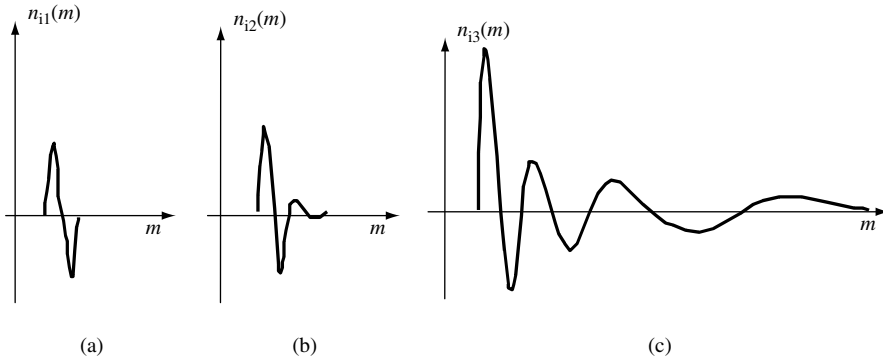


Figure 12.3 Illustration of variations of the impulse response of a nonlinear system with increasing amplitude of the impulse.

the channel characteristics with the amplitude of impulsive noise. These variations may be attributed to the nonlinear characteristics of the playback mechanism.

An important consideration in the development of a noise processing system is the choice of an appropriate domain (time or the frequency) for signal representation. The choice should depend on the specific objective of the system. In signal restoration, the objective is to separate the noise from the signal, and the representation domain must be the one that emphasizes the distinguishing features of the signal and the noise. Impulsive noise is normally more distinct and detectable in the time domain than in the frequency domain, and it is appropriate to use time-domain signal processing for noise detection and removal. In signal classification and parameter estimation, the objective may be to compensate for the average effects of the noise over a number of samples, and in some cases, it may be more appropriate to process the impulsive noise in the frequency domain where the effect of noise is a change in the mean of the power spectrum of the signal.

12.1.1 AUTOCORRELATION AND POWER SPECTRUM OF IMPULSIVE NOISE

Impulsive noise is a nonstationary, binary-state sequence of impulses with random amplitudes and random positions of occurrence. The nonstationary nature of impulsive noise can be seen by considering the power spectrum of a noise process with a few impulses per second: when the noise is absent the process has zero power, and when an impulse is present the noise power is the power of the impulse. Therefore the power spectrum and hence the autocorrelation of an impulsive noise is a binary state, time-varying process. An impulsive noise sequence can be modelled as an amplitude-modulated binary-state sequence, and expressed as

$$n_i(m) = n(m)b(m) \quad (12.6)$$

where $b(m)$ is a binary-state random sequence of ones and zeros, and $n(m)$ is a random noise process. Assuming that impulsive noise is an uncorrelated random process, the autocorrelation

of impulsive noise may be defined as a binary-state process:

$$r_{nn}(k, m) = \mathcal{E}[n_i(m)n_i(m+k)] = \sigma_n^2\delta(k)b(m) \tag{12.7}$$

where $\delta(k)$ is the Kronecker delta function. Since it is assumed that the noise is an uncorrelated process, the autocorrelation is zero for $k \neq 0$, therefore Equation (12.7) may be written as

$$r_{nn}(0, m) = \sigma_n^2b(m) \tag{12.8}$$

Note that for a zero-mean noise process, $r_{nn}(0, m)$ is the time-varying binary-state noise power. The power spectrum of an impulsive noise sequence is obtained by taking the Fourier transform of the autocorrelation function Equation (12.8), as

$$P_{N_iN_i}(f, m) = \sigma_n^2b(m) \tag{12.9}$$

In Equations (12.8) and (12.9), the autocorrelation and power spectrum are expressed as binary state functions that depend on the ‘on/off’ state of impulsive noise at time m .

12.2 STATISTICAL MODELS FOR IMPULSIVE NOISE

In this section, we study a number of statistical models for the characterisation of an impulsive noise process. An impulsive noise sequence, $n_i(m)$, consists of short duration pulses of a random amplitude, duration and time of occurrence, and may be modelled as the output of a filter excited by an amplitude-modulated random binary sequence as

$$n_i(m) = \sum_{k=0}^{P-1} h_k n(m-k)b(m-k) \tag{12.10}$$

Figure 12.4 illustrates the impulsive noise model of Equation (12.10). In Equation (12.10), $b(m)$ is a binary-valued random sequence model of the time of occurrence of impulsive

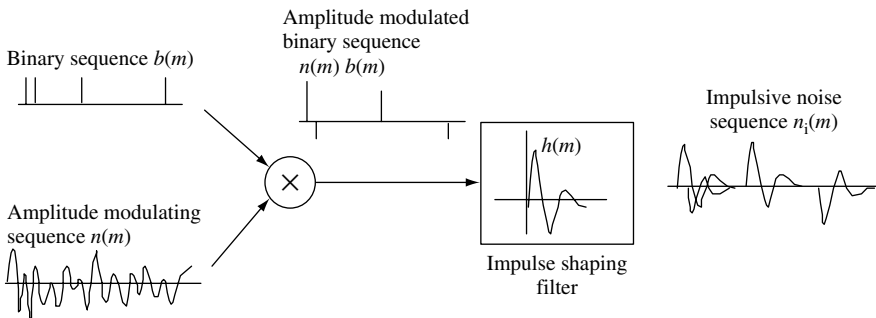


Figure 12.4 Illustration of an impulsive noise model as the output of a filter excited by an amplitude-modulated binary sequence.

noise, $n(m)$ is a continuous-valued random process model of impulse amplitude and $h(m)$ is the impulse response of a filter that models the duration and shape of each impulse. Two important statistical processes for modelling impulsive noise as an amplitude-modulated binary sequence are the Bernoulli–Gaussian process and the Poisson–Gaussian process, which are discussed next.

12.2.1 BERNOULLI–GAUSSIAN MODEL OF IMPULSIVE NOISE

In a Bernoulli–Gaussian model of an impulsive noise process, the random time of occurrence of the impulses is modelled by a binary Bernoulli process, $b(m)$, and the amplitude of the impulses is modelled by a Gaussian process, $n(m)$. A Bernoulli process, $b(m)$, is a binary-valued process that takes a value of ‘1’ with a probability of α and a value of ‘0’ with a probability of $1 - \alpha$. The probability mass function of a Bernoulli process is given by

$$P_B [b(m)] = \begin{cases} \alpha & \text{for } b(m) = 1 \\ 1 - \alpha & \text{for } b(m) = 0 \end{cases} \quad (12.11)$$

A Bernoulli process has a mean,

$$\mu_b = \mathcal{E} [b(m)] = \alpha \quad (12.12)$$

and a variance

$$\sigma_b^2 = \mathcal{E} \left\{ [b(m) - \mu_b]^2 \right\} = \alpha(1 - \alpha) \quad (12.13)$$

A zero-mean Gaussian pdf model of the random amplitudes of impulsive noise is given by

$$f_N [n(m)] = \frac{1}{\sqrt{2\pi}\sigma_n} \exp \left[-\frac{n^2(m)}{2\sigma_n^2} \right] \quad (12.14)$$

where σ_n^2 is the variance of the noise amplitude. In a Bernoulli–Gaussian model the probability density function of an impulsive noise, $n_i(m)$, is given by

$$f_N^{\text{BG}} [n_i(m)] = (1 - \alpha)\delta [n_i(m)] + \alpha f_N [n_i(m)] \quad (12.15)$$

where $\delta [n_i(m)]$ is the Kronecker delta function. Note that the function $f_N^{\text{BG}} [n_i(m)]$ is a mixture of a discrete probability mass function $\delta [n_i(m)]$ and a continuous probability density function $f_N [n_i(m)]$.

An alternative model for impulsive noise is a binary-state Gaussian process (Section 2.5.4), with a low-variance state modelling the absence of impulses and a relatively high-variance state modelling the amplitude of impulsive noise.

12.2.2 POISSON–GAUSSIAN MODEL OF IMPULSIVE NOISE

In a Poisson–Gaussian model the probability of occurrence of an impulsive noise event is modelled by a Poisson process, and the distribution of the random amplitude of impulsive

noise is modelled by a Gaussian process. The Poisson process, described in Chapter 2, is a random event-counting process. In a Poisson model, the probability of occurrence of k impulsive noise in a time interval T is given by

$$P(k, T) = \frac{(\lambda T)^k}{k!} e^{-\lambda T} \tag{12.16}$$

where λ is a rate function with the following properties:

$$\begin{aligned} \text{Prob}(\text{one impulse in a small time interval } \Delta t) &= \lambda \Delta t \\ \text{Prob}(\text{zero impulse in a small time interval } \Delta t) &= 1 - \lambda \Delta t \end{aligned} \tag{12.17}$$

It is assumed that no more than one impulsive noise can occur in a time interval Δt . In a Poisson–Gaussian model, the pdf of an impulsive noise, $n_i(m)$, in a small time interval Δt is given by

$$f_{N_i}^{\text{PG}} [n_i(m)] = (1 - \lambda \Delta t) \delta [n_i(m)] + \lambda \Delta t f_N [n_i(m)] \tag{12.18}$$

where $f_N [n_i(m)]$ is the Gaussian pdf of Equation (12.14).

12.2.3 A BINARY-STATE MODEL OF IMPULSIVE NOISE

An impulsive noise process may be modelled by a binary-state model, as shown in Figure 12.5. In this binary model, the state S_0 corresponds to the ‘off’ condition when impulsive noise is absent; in this state, the model emits zero-valued samples. The state S_1 corresponds to the ‘on’ condition; in this state the model emits short-duration pulses of random amplitude and duration. The probability of a transition from state S_i to state S_j is denoted by a_{ij} . In its simplest form, as shown in Figure 12.5, the model is memoryless, and the probability of a transition to state S_i is independent of the current state of the model. In this case, the probability that at time $t + 1$ the signal is in the state S_0 is independent of the state at time t , and is given by

$$P [s(t + 1) = S_0 | s(t) = S_0] = P [s(t + 1) = S_1 | s(t) = S_1] = 1 - \alpha \tag{12.19}$$

where s_t denotes the state at time t . Likewise, the probability that at time $t + 1$ the model is in state S_1 is given by

$$P [s(t + 1) = S_1 | s(t) = S_0] = P [s(t + 1) = S_1 | s(t) = S_1] = \alpha \tag{12.20}$$

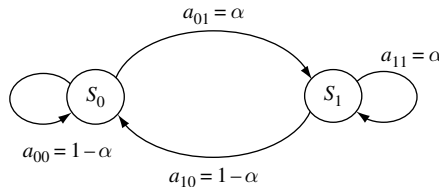


Figure 12.5 A binary-state model of an impulsive noise generator.

In a more general form of the binary-state model, a Markovian state-transition can model the dependencies in the noise process. The model then becomes a two-state hidden Markov model, as considered in Chapter 5.

In one of its simplest forms, the state S_1 emits samples from a zero-mean Gaussian random process. The impulsive noise model in state S_1 can be configured to accommodate a variety of impulsive noise of different shapes, durations and pdfs. A practical method for modelling a variety of impulsive noise is to use a codebook of M prototype impulsive noises, and their associated probabilities $[(n_{i1}, p_{i1}), (n_{i2}, p_{i2}), \dots, (n_{iM}, p_{iM})]$, where p_j denotes the probability of impulsive noise of the type n_j . The impulsive noise code book may be designed by classification of a large number of ‘training’ impulsive noises into a relatively small number of clusters. For each cluster, the average impulsive noise is chosen as the representative of the cluster. The number of impulses in the cluster of type j divided by the total number of impulses in all clusters gives p_j , the probability of an impulse of type j .

Figure 12.6 shows a three-state model of the impulsive noise and the decaying oscillations that might follow the noise. In this model, the state S_0 models the absence of impulsive noise, the state S_1 models the impulsive noise and the state S_2 models any oscillations that may follow a noise pulse.

12.2.4 SIGNAL-TO-IMPULSIVE-NOISE RATIO

For impulsive noise the average signal-to-impulsive-noise ratio, averaged over an entire noise sequence including the time instances when the impulses are absent, depends on two parameters: (a) the average power of each impulsive noise; and (b) the rate of occurrence of impulsive noise. Let P_{impulse} denote the average power of each impulse, and P_{signal} the signal power. We may define a ‘local’ time-varying signal-to-impulsive-noise ratio as

$$\text{SINR}(m) = \frac{P_{\text{Signal}}(m)}{P_{\text{impulse}} b(m)} \quad (12.21)$$

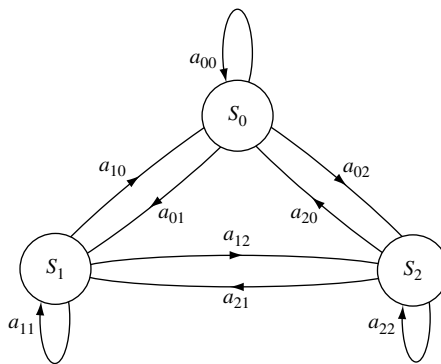


Figure 12.6 A three-state model of impulsive noise and the decaying oscillations that often follow the impulses.

The average signal-to-impulsive-noise ratio, assuming that the parameter α is the fraction of signal samples contaminated by impulsive noise, can be defined as

$$\text{SINR} = \frac{P_{\text{Signal}}}{\alpha P_{\text{Impulse}}} \tag{12.22}$$

Note that from Equation (12.22), for a given signal power, there are many pair of values of α and P_{impulse} that can yield the same average SINR.

12.3 MEDIAN FILTERS

The classical approach to removal of impulsive noise is the median filter. The median of a set of samples $\{x(m)\}$ is a member of the set $x_{\text{med}}(m)$ such that half the population of the set is larger than $x_{\text{med}}(m)$ and half is smaller than $x_{\text{med}}(m)$. Hence the median of a set of samples is obtained by sorting the samples in ascending or descending order, and then selecting the mid-value. In median filtering, a window of predetermined length slides sequentially over the signal, and the mid-sample within the window is replaced by the median of all the samples that are inside the window, as illustrated in Figure 12.7.

The output $\hat{x}(m)$ of a median filter with input $y(m)$ and a median window of length $2K + 1$ samples is given by

$$\begin{aligned} \hat{x}(m) &= y_{\text{med}}(m) \\ &= \text{median}[y(m - K), \dots, y(m), \dots, y(m + K)] \end{aligned} \tag{12.23}$$

The median of a set of numbers is a nonlinear statistic of the set, with the useful property that it is insensitive to the presence of a sample with an unusually large value, an ‘outlier’,

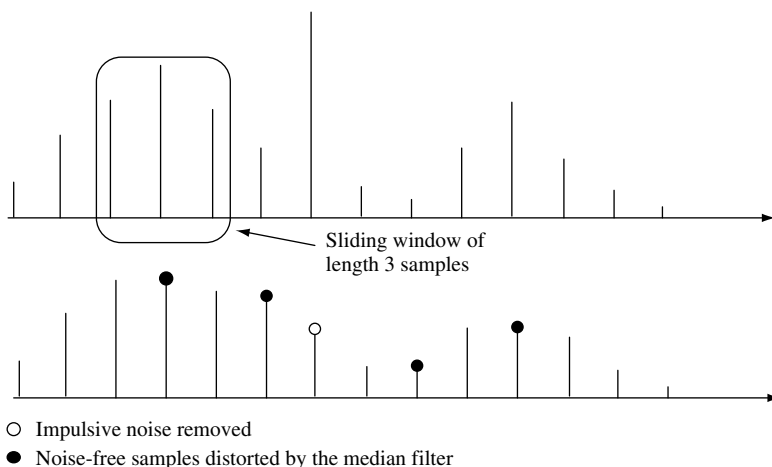


Figure 12.7 Input and output of a median filter. Note that, in addition to suppressing the impulsive outlier, the filter also distorts some genuine signal components.

in the set. In contrast, the mean, and, in particular, the variance of a set of numbers are sensitive to the presence of impulsive-type noise. An important property of median filters, particularly useful in image processing, is that they preserve edges or stepwise discontinuities in the signal. Median filters can be used for removing impulses in an image without smearing the edge information; this is of significant importance in image processing. However, experiments with median filters for removal of impulsive noise from audio signals demonstrate that median filters are unable to produce high-quality audio restoration. The median filters cannot deal with ‘real’ impulsive noise, which is often more than one or two samples long. Furthermore, median filters introduce a great deal of processing distortion by modifying genuine signal samples that are mistaken for impulsive noise. The performance of median filters may be improved by employing an adaptive threshold, so that a sample is replaced by the median only if the difference between the sample and the median is above the threshold:

$$\hat{x}(m) = \begin{cases} y(m) & \text{if } |y(m) - y_{\text{med}}(m)| < k\theta(m) \\ y_{\text{med}}(m) & \text{otherwise} \end{cases} \quad (12.24)$$

where $\theta(m)$ is an adaptive threshold that may be related to a robust estimate of the average of $|y(m) - y_{\text{med}}(m)|$, and k is a tuning parameter. Median filters are not optimal, because they do not make efficient use of prior knowledge of the physiology of signal generation, or a model of the signal and noise statistical distributions. In the following section we describe an autoregressive model-based impulsive removal system, capable of producing high-quality audio restoration.

12.4 IMPULSIVE NOISE REMOVAL USING LINEAR PREDICTION MODELS

In this section, we study a model-based impulsive noise removal system. Impulsive disturbances usually contaminate a relatively small fraction α of the total samples. Since a large fraction, $1 - \alpha$, of samples remains unaffected by impulsive noise, it is advantageous to locate individual noise pulses, and correct *only* those samples that are distorted. This strategy avoids the unnecessary processing and compromise in the quality of the relatively large fraction of samples that is not disturbed by impulsive noise. The impulsive noise removal system shown in Figure 12.8 consists of two subsystems: a detector and an interpolator. The detector locates the position of each noise pulse, and the interpolator replaces the distorted samples using the samples on both sides of the impulsive noise. The detector is composed of a linear prediction analysis system, a matched filter and a threshold detector. The output of the detector is a binary switch and controls the interpolator. A detector output of ‘0’ signals the absence of impulsive noise and the interpolator is bypassed. A detector output of ‘1’ signals the presence of impulsive noise, and the interpolator is activated to replace the samples obliterated by noise.

12.4.1 IMPULSIVE NOISE DETECTION

A simple method for detection of impulsive noise is to employ an amplitude threshold and classify those samples with amplitudes above the threshold, as noise. This method

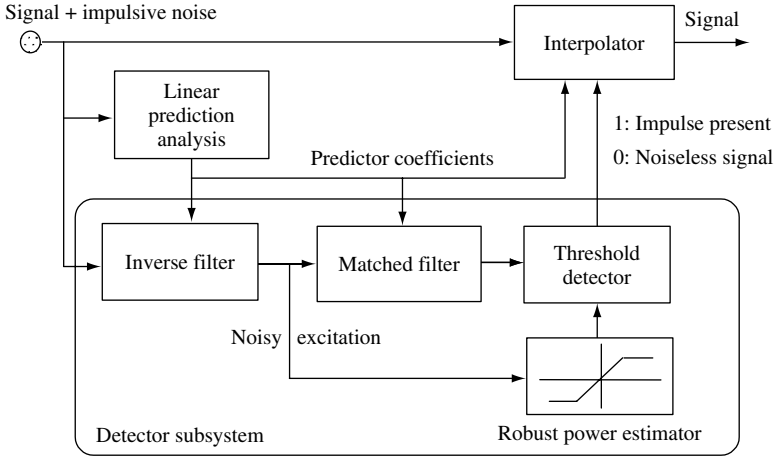


Figure 12.8 Configuration of an impulsive noise-removal system incorporating a detector and interpolator subsystems.

works fairly well for relatively large-amplitude impulses, but fails when the noise amplitude falls below the signal. Detection can be improved by utilising the characteristic differences between the impulsive noise and the signal. An impulsive noise, or a short-duration pulse, introduces uncharacteristic discontinuity in a correlated signal. The discontinuity becomes more detectable when the signal is differentiated. The differentiation (or, for digital signals, the differencing) operation is equivalent to decorrelation or spectral whitening. In this section, we describe a model-based decorrelation method for improving impulsive noise detectability. The correlation structure of the signal is modelled by a linear predictor, and the process of decorrelation is achieved by inverse filtering. Linear prediction and inverse filtering are covered in Chapter 8. Figure 12.9 shows a model for a noisy signal. The noise-free signal, $x(m)$, is described by a linear prediction model as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \tag{12.25}$$

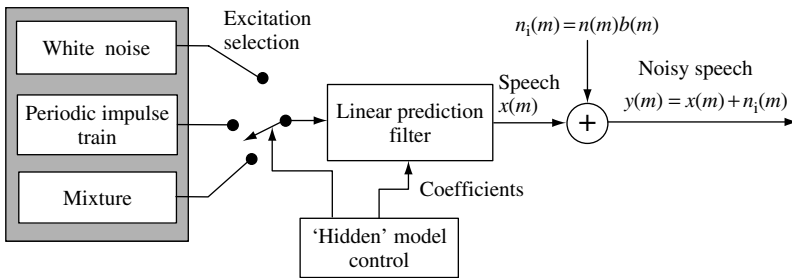


Figure 12.9 Noisy speech model. The signal is modelled by a linear predictor. Impulsive noise is modelled as an amplitude-modulated binary-state process.

where $\mathbf{a} = [a_1, a_2, \dots, a_P]^T$ is the coefficient vector of a linear predictor of order P , and the excitation $e(m)$ is either a noise-like signal or a mixture of a random noise and a quasiperiodic train of pulses, as illustrated in Figure 12.9. The impulsive noise detector is based on the observation that linear predictors are a good model of the correlated signals but not the uncorrelated binary-state impulsive-type noise. Transforming the noisy signal, $y(m)$, to the excitation signal of the predictor has the following effects:

- (1) The scale of the signal amplitude is reduced to almost that of the original excitation signal, whereas the scale of the noise amplitude remains unchanged or increases.
- (2) The signal is decorrelated, whereas the impulsive noise is smeared and transformed to a scaled version of the impulse response of the inverse filter.

Both effects improve noise detectability. Speech or music is composed of random excitations spectrally shaped and amplified by the resonances of the vocal tract or the musical instruments. The excitation is more random than the speech, and often has a much smaller amplitude range. The improvement in noise pulse detectability obtained by inverse filtering can be substantial and depends on the time-varying correlation structure of the signal. Note that this method effectively reduces the impulsive noise detection to the problem of separation of outliers from a random noise excitation signal using some optimal thresholding device.

12.4.2 ANALYSIS OF IMPROVEMENT IN NOISE DETECTABILITY

In the following, the improvement in noise detectability that results from inverse filtering is analysed. Using Equation (12.25), we can rewrite a noisy signal model as

$$\begin{aligned} y(m) &= x(m) + n_i(m) \\ &= \sum_{k=1}^P a_k x(m-k) + e(m) + n_i(m) \end{aligned} \quad (12.26)$$

where $y(m)$, $x(m)$ and $n_i(m)$ are the noisy signal, the signal and the noise respectively. Using an estimate $\hat{\mathbf{a}}$ of the predictor coefficient vector \mathbf{a} , the noisy signal, $y(m)$, can be inverse-filtered and transformed to the noisy excitation signal, $v(m)$, as

$$\begin{aligned} v(m) &= y(m) - \sum_{k=1}^P \hat{a}_k y(m-k) \\ &= x(m) + n_i(m) - \sum_{k=1}^P (a_k - \tilde{a}_k) [x(m-k) + n_i(m-k)] \end{aligned} \quad (12.27)$$

where \tilde{a}_k is the error in the estimate of the predictor coefficient. Using Equation (12.25), Equation (12.27) can be rewritten in the following form:

$$v(m) = e(m) + n_i(m) + \sum_{k=1}^P \tilde{a}_k x(m-k) - \sum_{k=1}^P \hat{a}_k n_i(m-k) \quad (12.28)$$

From Equation (12.28) there are essentially three terms that contribute to the noise in the excitation sequence:

- (1) the impulsive disturbance $n_i(m)$, which is usually the dominant term;
- (2) the effect of the past P noise samples, smeared to the present time by the action of the inverse filtering, $\sum \hat{a}_k n_i(m - k)$;
- (3) the increase in the variance of the excitation signal, caused by the error in the parameter vector estimate, and expressed by the term $\sum \tilde{a}_k x(m - k)$.

The improvement resulting from the inverse filter can be formulated as follows. The impulsive noise to signal ratio for the noisy signal is given by

$$\frac{\text{impulsive noise power}}{\text{signal power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[x^2(m)]} \tag{12.29}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. Note that, in impulsive noise detection, the signal of interest is the impulsive noise to be detected from the accompanying signal. Assuming that the dominant noise term in the noisy excitation signal, $v(m)$, is the impulse $n_i(m)$, the impulsive noise-to-excitation-signal ratio is given by

$$\frac{\text{impulsive noise power}}{\text{excitation power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[e^2(m)]} \tag{12.30}$$

The overall gain in impulsive-noise-to-signal ratio is obtained by dividing Equations (12.29) and (12.30), as

$$\frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]} = \text{gain} \tag{12.31}$$

This simple analysis demonstrates that the improvement in impulsive noise detectability depends on the power amplification characteristics, due to resonances, of the linear predictor model. For speech signals, the scale of the amplitude of the noiseless speech excitation is on the order of 10^{-1} to 10^{-4} that of the speech itself; therefore substantial improvement in impulsive noise detectability can be expected through inverse filtering of the noisy speech signals.

Figure 12.10 illustrates the effect of inverse filtering in improving the detectability of impulsive noise. The inverse filtering has the effect that the signal $x(m)$ is transformed to an uncorrelated excitation signal, $e(m)$, whereas the impulsive noise is smeared to a scaled version of the inverse filter impulse response $[1, -a_1, \dots, -a_p]$, as indicated by the term $\sum \hat{a}_k n_i(m - k)$ in Equation (12.28). Assuming that the excitation is a white noise Gaussian signal, a filter matched to the inverse filter coefficients may enhance the delectability of the smeared impulsive noise from the excitation signal.

12.4.3 TWO-SIDED PREDICTOR FOR IMPULSIVE NOISE DETECTION

In the previous section, it was shown that impulsive noise detectability can be improved by decorrelating the speech signal. The process of decorrelation can be taken further by the use

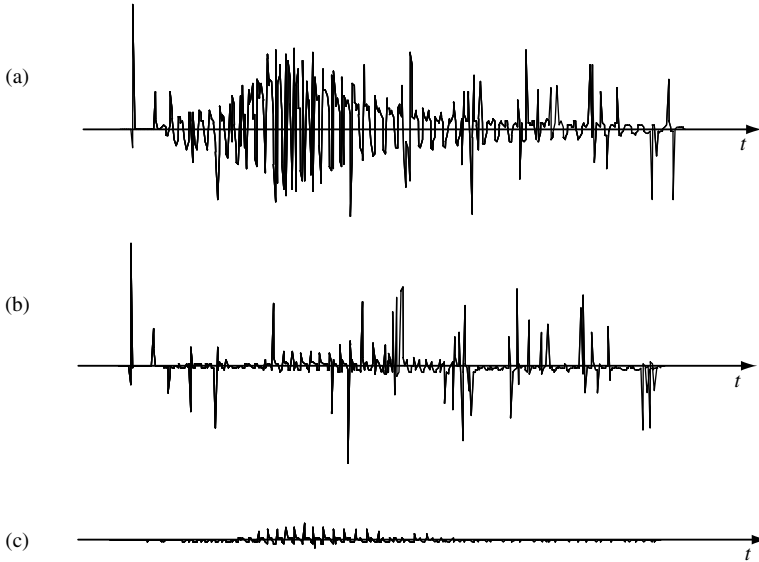


Figure 12.10 Illustration of the effects of inverse filtering on detectability of impulsive noise: (a) impulsive noise contaminated speech with 5% impulse contamination at an average SINR of 10 dB; (b) speech excitation of impulse contaminated speech; and (c) speech excitation of impulse-free speech.

of a two-sided linear prediction model. The two-sided linear prediction of a sample $x(m)$ is based on the P past samples and the P future samples, and is defined by the equation

$$x(m) = \sum_{k=1}^P a_k x(m-k) + \sum_{k=1}^P a_{k+p} x(m+k) + e(m) \quad (12.32)$$

where a_k is the two-sided predictor coefficient and $e(m)$ is the excitation signal. All the analysis used for the case of a one-sided linear predictor can be extended to the two-sided model. However, the variance of the excitation input of a two-sided model is less than that of the one-sided predictor because in Equation (12.32) the correlations of each sample with the future, as well as the past, samples are modelled. Although Equation (12.32) is a noncausal filter, its inverse, required in the detection subsystem, is causal. The use of a two-sided predictor can result in further improvement in noise detectability.

12.4.4 INTERPOLATION OF DISCARDED SAMPLES

Samples irrevocably distorted by an impulsive noise are discarded and the gap thus left is interpolated. For interpolation imperfections to remain inaudible, a high-fidelity interpolator is required. A number of interpolators for replacement of a sequence of missing samples were introduced in Chapter 10. The least square autoregressive (LSAR) interpolation algorithm of Section 10.3.2 produces high-quality results for a relatively small number of missing samples left by an impulsive noise. The LSAR interpolation method is a two-stage process.

In the first stage, the available samples on both sides of the noise pulse are used to estimate the parameters of a linear prediction model of the signal. In the second stage, the estimated model parameters and the samples on both sides of the gap are used to interpolate the missing samples. The use of this interpolator in replacement of audio signals distorted by impulsive noise has produced high-quality results.

12.5 ROBUST PARAMETER ESTIMATION

In Figure 12.8, the threshold used for detection of impulsive noise from the excitation signal is derived from a nonlinear robust estimate of the excitation power. In this section, we consider the robust estimation of a parameter, such as the signal power, in the presence of impulsive noise.

A *robust* estimator is one that is not oversensitive to deviations of the input signal from the assumed distribution. In a robust estimator, an input sample with unusually large amplitude has only a limited effect on the estimation results. Most signal processing algorithms developed for adaptive filtering, speech recognition, speech coding, etc., are based on the assumption that the signal and the noise are Gaussian-distributed, and employ a mean square distance measure as the optimality criterion. The mean square error criterion is sensitive to non-Gaussian events such as impulsive noise. A large impulsive noise in a signal can substantially overshadow the influence of noise-free samples.

Figure 12.11 illustrates the variations of several cost-of-error functions with a parameter θ . Figure 12.11(a) shows a least square error cost function and its influence function. The influence function is the derivative of the cost function and, as the name implies, it has a direct influence on the estimation results. It can be seen from the influence function of Figure 12.11(a) that an unbounded sample has an unbounded influence on the estimation results.

A method for introducing robustness is to use a nonlinear function and limit the influence of any one sample on the overall estimation results. The absolute value of error is a robust cost function, as shown by the influence function in Figure 12.11(b). One disadvantage of this function is that it is not continuous at the origin. A further drawback is that it does not allow for the fact that, in practice, a large proportion of the samples is not contaminated with impulsive noise, and may well be modelled with Gaussian densities.

Many processes may be regarded as Gaussian for the sample values that cluster about the mean. For such processes, it is desirable to have an influence function that limits the influence of outliers and at the same time is linear and optimal for the large number of relatively small-amplitude samples that may be regarded as Gaussian-distributed. One such function is Huber's function, defined as

$$\psi[e(m)] = \begin{cases} e^2(m), & \text{if } |e(m)| \leq k \\ k |e(m)| & \text{otherwise} \end{cases} \tag{12.33}$$

Huber's function, shown in Figure 12.11(c), is a hybrid of the least mean square and the absolute value of error functions. Tukey's bi-weight function, which is a redescending robust objective function, is defined as

$$\psi[e(m)] = \begin{cases} \{1 - [1 - e^2(m)]^3\}/6 & \text{if } |e(m)| \leq 1 \\ 1/6 & \text{otherwise} \end{cases} \tag{12.34}$$

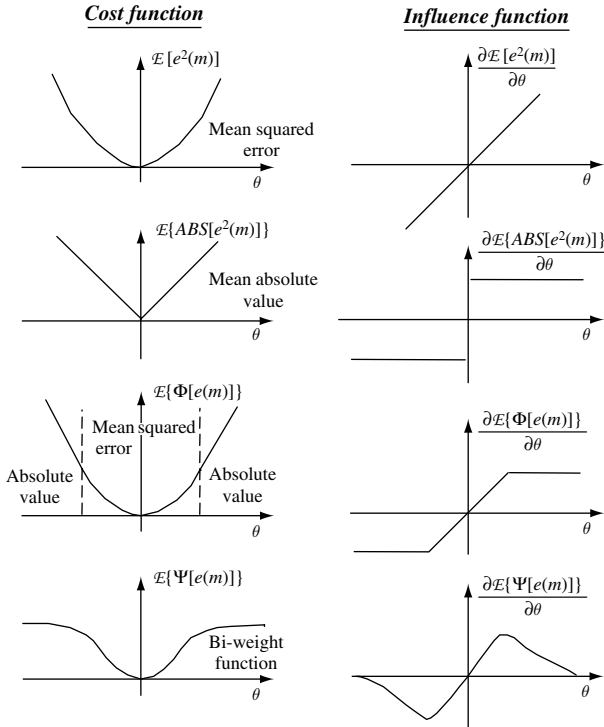


Figure 12.11 Illustration of a number of cost-of-error functions and the corresponding influence functions.

As shown in Figure 12.11(d), the influence function is linear for small signal values but introduces attenuation as the signal value exceeds some threshold. The threshold may be obtained from a robust median estimate of the signal power.

12.6 RESTORATION OF ARCHIVED GRAMOPHONE RECORDS

This section describes the application of the impulsive noise removal system of Figure 12.8 to the restoration of archived audio records. As the bandwidth of archived recordings is limited to 7–8 kHz, a low-pass, anti-aliasing filter with a cutoff frequency of 8 kHz is used to remove the out-of-band noise. Played back signals were sampled at a rate of 20 kHz, and digitised to 16 bits. Figure 12.12(a) shows a 25 ms segment of noisy music and song from an old 78 rpm gramophone record. The impulsive interferences are due to faults in the record stamping process, granularities of the record material or physical damage. This signal is modelled by a predictor of order 20. The excitation signals obtained from the inverse filter and the matched filter output are shown in Figure 12.12(b) and (c), respectively. Close examination of these figures shows that some of the ambiguities between the noise pulses and the genuine signal excitation pulses are resolved after matched filtering.

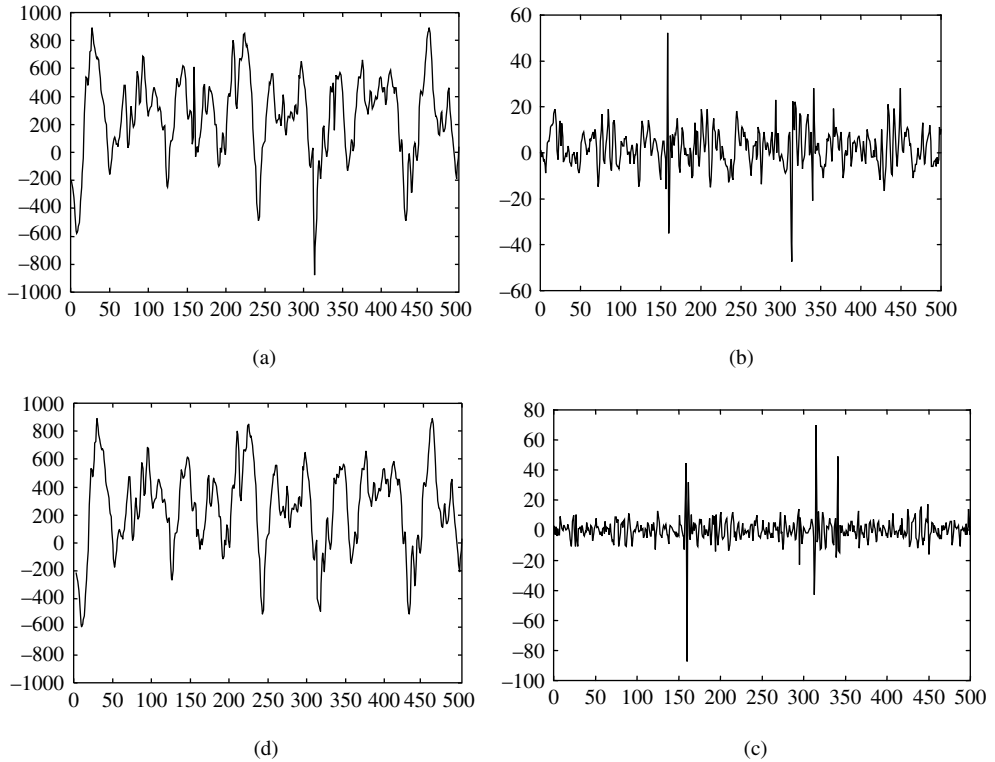


Figure 12.12 (a) A noisy audio signal from a 78 rpm record; (b) noisy excitation signal; (c) matched filter output; (d) restored signal.

The amplitude threshold for detection of impulsive noise from the excitation signal is adapted on a block basis, and is set to $k\sigma_e^2$, where σ_e^2 is a robust estimate of the excitation power. The robust estimate is obtained by passing the noisy excitation signal through a soft nonlinearity that rejects outliers. The scalar k is a tuning parameter; the choice of k reflects a trade-off between the hit rate and the false-alarm rate of the detector. As k decreases, smaller noise pulses are detected but the false detection rate also increases. When an impulse is detected, a few samples are discarded and replaced by the LSAR interpolation algorithm described in Chapter 10. Figure 12.12(d) shows the signal with the impulses removed. The impulsive noise removal system of Figure 12.8 was successfully applied to restoration of numerous examples of archived gramophone records. The system is also effective in suppressing impulsive noise in examples of noisy telephone conversations.

12.7 SUMMARY

The classic linear time-invariant theory on which many signal processing methods are based is not suitable for dealing with the nonstationary impulsive noise problem. In this chapter, we considered impulsive noise as a random on/off process and studied several stochastic models

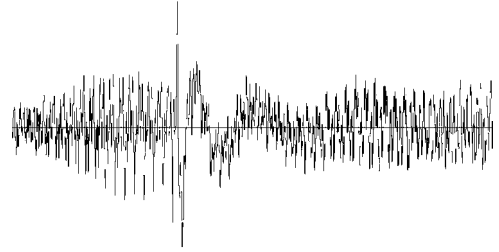
for impulsive noise, including the Bernoulli–Gaussian, the Poisson–Gaussian and the hidden Markov models. The HMM provides a particularly interesting framework, because the theory of HMM studied in Chapter 5 is well developed, and also because the state sequence of an HMM of noise can be used to provide an estimate of the presence or the absence of the noise. By definition, an impulsive noise is a short and sharp event uncharacteristic of the signal that it contaminates. In general, differencing enhances the detectibility of impulsive noise. Based on this observation, in Section 12.4 we considered an algorithm based on a linear prediction model of the signal for detection of impulsive noise.

In the next chapter we expand the materials we considered here for the modelling, detection and removal of transient noise pulses.

BIBLIOGRAPHY

- Dempster A.P., Laird N.M and Rubin D.B. (1971) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc.*, **39**: 1–38.
- Gallagher N.C. and Wise G.L. (1981) A theoretical analysis of the properties of median filters. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-29**: 1136–1141.
- Godsil S. (1993) *Restoration of Degraded Audio Signals*. Cambridge University Press, Cambridge.
- Jayant N.S. (1976) Average and median based smoothing for improving digital speech quality in the presence of transmission errors. *IEEE Trans. Commun.*, **24**(9): 1043–1045.
- Kelma V.C and Laub A.J (1980) The singular value decomposition: its computation and some applications. *IEEE Trans. Autom. Control*, **AC-25**: 164–176.
- Kunda A., Mitra S. and Vaidyanathan P. (1984) Applications of two dimensional generalised mean filtering for removal of impulsive noise from images. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-32**(3): 600–609.
- Nieminen, Heinonen P. and Neuvo Y. (1987) Suppression and detection of impulsive type interference using adaptive median hybrid filters. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-87*, pp. 117–120.
- Rabiner L.R., Sambur M.R. and Schmidt C.E. (1984) Applications of a nonlinear smoothing algorithm to speech processing. *IEEE Trans. Speech Signal Process.*, **ASSP-23**(6): 552–557.
- Tukey J.W. (1971) *Exploratory Data Analysis*. Addison Wesley, Reading, MA.
- Vaseghi S.V. and Milner B.P. (1995) Speech recognition in impulsive noise. *IEEE Proceedings International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, pp. 437–440.
- Vaseghi S.V. and Rayner P.J.W (1990) Detection and suppression of impulsive noise in speech communication systems. *IEE Proceedings I: Communications Speech and Vision*, February, pp. 38–46.

13



Transient Noise Pulses

13.1 Transient Noise Waveforms	337
13.2 Transient Noise Pulse Models	339
13.3 Detection of Noise Pulses	342
13.4 Removal of Noise Pulse Distortions	345
13.5 Summary	349
Bibliography	349

Transient noise pulses differ from the short-duration impulsive noise studied in the previous chapter, in that they have a longer duration and a relatively higher proportion of low-frequency energy content, and usually occur less frequently than impulsive noise. The sources of transient noise pulses are varied, and may be electromagnetic, acoustic or due to physical defects in the recording medium. Examples of transient noise pulses include switching noise in telephony, noise pulses due to adverse radio transmission environments, noise pulses due to on/off switching of nearby electric devices, scratches and defects on damaged records, click sounds from a computer keyboard, etc. The noise pulse removal methods considered in this chapter are based on the observation that transient noise pulses can be regarded as the response of the communication channel, or the playback system, to an impulse. In this chapter, we study the characteristics of transient noise pulses and consider a template-based method, a linear predictive model and a hidden Markov model for the modelling and removal of transient noise pulses. The subject of this chapter closely follows that of Chapter 12 on impulsive noise.

13.1 TRANSIENT NOISE WAVEFORMS

Transient noise pulses often consist of a relatively short sharp initial pulse followed by decaying low-frequency oscillations, as shown in Figure 13.1. The initial pulse is usually

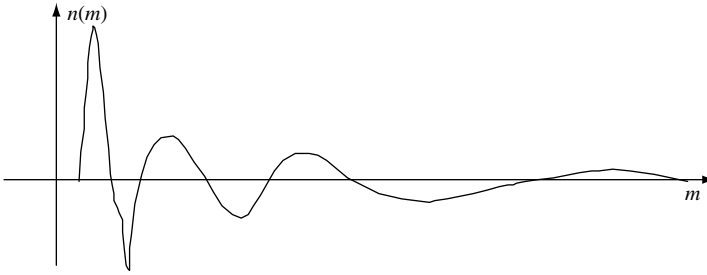


Figure 13.1 The profile of a transient noise pulse from a scratched gramophone record.

due to some external or internal impulsive interference, whereas the oscillations are often due to the resonance of the communication channel excited by the initial pulse, and may be considered as the response of the channel to the initial pulse. In a telecommunications system, a noise pulse originates at some point in time and space, and then propagates through the channel to the receiver. The noise pulse is shaped by the channel characteristics, and may be considered as the channel pulse response. Thus we expect to be able to characterize the transient noise pulses with a similar degree of consistency to that of the channels through which the pulses propagate.

As an illustration of the distribution of a transient noise pulse in time and frequency, consider the scratch pulses from a damaged gramophone record shown in Figures 13.1 and 13.2. Scratch noise pulses are acoustic manifestations of the response of the stylus and the associated electromechanical playback system to a sharp physical discontinuity on the recording medium. Since scratches are essentially the impulse response of the playback mechanism, it is expected that, for a given system, various scratch pulses exhibit a similar characteristics. As shown in Figure 13.1, a typical scratch waveform often exhibits two distinct regions:

- (1) the initial high-amplitude pulse response of the playback system to the physical discontinuity on the record medium; this is followed by
- (2) decaying oscillations that cause additive distortion.

The initial pulse is relatively short and has a duration on the order of 1–5 ms, whereas the oscillatory tail has a longer duration and may last up to 50 ms. Note in Figure 13.1 that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the nonlinear modes of response of the electromechanical playback system excited by the physical scratch discontinuity. Observations of many scratch waveforms from damaged gramophone records reveal that they have a well-defined profile, and can be characterised by a relatively small number of typical templates.

A similar argument can be used to describe the transient noise pulses in other systems as the response of the system to an impulsive noise. Figure 13.2(a) and (b) shows the time-domain waveform and the spectrogram of a section of music and song with scratch-type noise. Note that, as the scratch defect on the record was radial, the scratch pulses occur periodically with a period of 78 pulses per scratch per minute. As can be seen, there were in fact two scratches on the record.

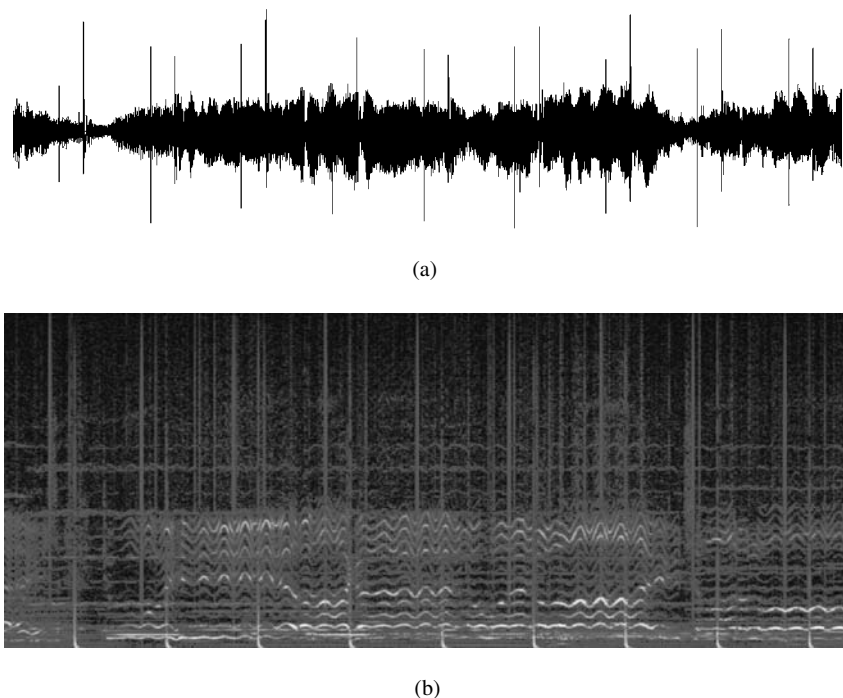


Figure 13.2 An example of (a) the time-domain waveform and (b) the spectrogram of transient noise scratch pulses in a damaged gramophone record.

The observation that transient noise pulses exhibit certain distinct, definable and consistent characteristics can be used for the modelling, detection and removal of transient noise pulses.

13.2 TRANSIENT NOISE PULSE MODELS

To a first approximation, a transient noise pulse, $n(m)$, can be modelled as the impulse response of a linear time-invariant filter model of the channel as

$$n(m) = \sum_k h_k A \delta(m - k) = A h_m \tag{13.1}$$

where A is the amplitude of the driving impulse and h_k is the channel impulse response. A burst of overlapping, or closely spaced, noise pulses can be modelled as the response of a channel to a sequence of impulses as

$$n(m) = \sum_k h_k \sum_j A_j \delta[(m - T_j) - k] = \sum_j A_j h_{m - T_j} \tag{13.2}$$

where it is assumed that the j th transient pulse is due to an impulse of amplitude A_j at time T_j . In practice, a noise model should be able to deal with the statistical variations of a

variety of noise and channel types. In this section, we consider three methods for modelling the temporal, spectral and durational characteristics of a transient noise pulse process:

- (1) a template-based model;
- (2) a linear-predictive model;
- (3) a hidden Markov model.

13.2.1 NOISE PULSE TEMPLATES

A widely used method for modelling the space of a random process is to model the process as a collection of signal clusters, and to design a codebook of templates containing the ‘centroids’ of the clusters. The centroids represent various typical forms of the process. To obtain the centroids, the signal space is partitioned into a number of regions or clusters, and the ‘centre’ of the space within each cluster is taken as a centroid of the signal process.

Similarly, a codebook of transient noise pulses can be designed by collecting a large number of training examples of the noise, and then using a clustering technique to group, or partition, the noise database into a number of clusters of noise pulses. The centre of each cluster is taken as a centroid of the noise space. Clustering techniques can be used to obtain a number of prototype templates for the characterisation of a set of transient noise pulses. The clustering of a noise process is based on a set of noise features that best characterise the noise. Features derived from the magnitude spectrum are commonly used for the characterisation of many random processes. For transient noise pulses, the most important features are the pulse shape, the temporal–spectral characteristics of the pulse, the pulse duration and the pulse energy profile. Figure 13.3 shows a number of typical noise pulses. The design of a codebook of signal templates is described in Chapter 4.

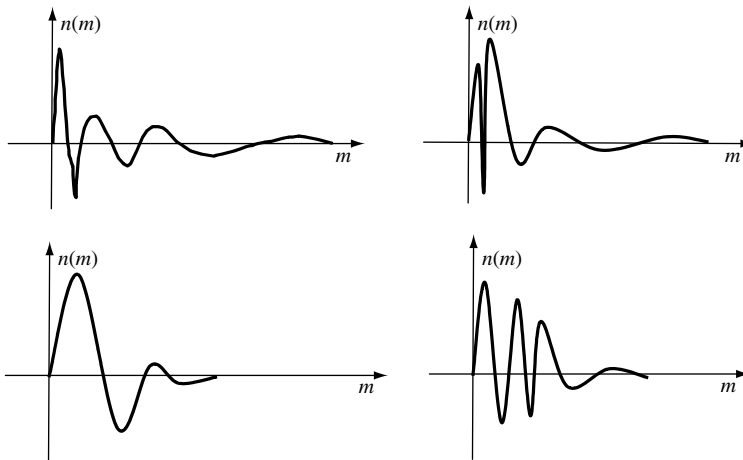


Figure 13.3 A number of prototype transient pulses.

13.2.2 AUTOREGRESSIVE MODEL OF TRANSIENT NOISE PULSES

Model-based methods have the advantage over template-based methods that overlapped noise pulses can be modelled as the response of the model to a number of closely spaced impulsive inputs. In this section, we consider an autoregressive model of transient noise pulses. The AR model for a single noise pulse $n(m)$ can be described as

$$n(m) = \sum_{k=1}^P c_k n(m-k) + A\delta(m) \quad (13.3)$$

where c_k is the AR model coefficient, and the excitation is an impulse function, $\delta(m)$, of amplitude A . A number of closely spaced and overlapping transient noise pulses can be modelled as the response of the AR model to a sequence of impulses:

$$n(m) = \sum_{k=1}^P c_k n(m-k) + \sum_j^M A_j \delta(m-T_j) \quad (13.4)$$

where it is assumed that T_j is the start of the j th pulse in a burst of M excitation pulses.

An improved AR model for transient noise, proposed by Godsill, is driven by a two-state excitation: in the state S_0 , the excitation is a zero-mean Gaussian process of small variance σ_0^2 , and in the state S_1 , the excitation is a zero-mean Gaussian process of relatively larger variance, $\sigma_1^2 \gg \sigma_0^2$. In the state S_1 a short-duration, and relatively large-amplitude, excitation generates a linear model of the transient noise pulse. In the state S_0 the model generates a low-amplitude excitation that partially models the inaccuracies of approximating a transient noise pulse by a linear predictive model. The binary-state excitation signal can be expressed as

$$e_n(m) = [\sigma_1 b(m) + \sigma_0 \bar{b}(m)] u(m) \quad (13.5)$$

where $u(m)$ is an uncorrelated zero-mean unit-variance Gaussian process and $b(m)$ indicates the state of the excitation signal: $b(m) = 1$ indicates that the excitation has a variance of σ_1^2 , and $b(m) = 0$ [or its binary complement $\bar{b}(m) = 1$] indicates the excitation has a smaller variance of σ_0^2 . The time-varying variance of $e_n(m)$ can be expressed as

$$\sigma_{e_n}^2(m) = \sigma_1^2 b(m) + \sigma_0^2 \bar{b}(m) \quad (13.6)$$

Assuming that the excitation pattern $b(m)$ is given, and that the excitation amplitude is Gaussian, the pdf of an N -sample long noise pulse, \mathbf{n} , is given by

$$f_N(\mathbf{n}) = \frac{1}{(2\pi)^{N/2} |\Lambda_{e_n e_n}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{n}^T \mathbf{C}^T \Lambda_{e_n e_n}^{-1} \mathbf{C} \mathbf{n}\right) \quad (13.7)$$

where \mathbf{C} is a matrix of coefficients of the AR model of the noise (as described in Section 8.4), and $\Lambda_{e_n e_n}$ is the diagonal covariance matrix of the input to the noise model. The diagonal elements of $\Lambda_{e_n e_n}$ are given by Equation (13.6).

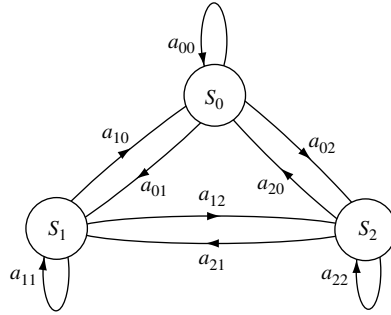


Figure 13.4 A three-state model of a transient noise pulse process.

13.2.3 HIDDEN MARKOV MODEL OF A NOISE PULSE PROCESS

A hidden Markov model, described in Chapter 5, is a finite state statistical model for nonstationary random processes such as speech or transient noise pulses. In general, we may identify three distinct states for a transient noise pulse process:

- (1) the periods during which there are no noise pulses;
- (2) the initial, and often short and sharp, pulse of a transient noise;
- (3) the decaying oscillatory tail of a transient pulse.

Figure 13.4 illustrates a three-state HMM of transient noise pulses. The state S_0 models the periods when the noise pulses are absent. In this state, the noise process may be zero-valued. This state can also be used to model a different noise process such as a white noise process. The state S_1 models the relatively sharp pulse that forms the initial part of many transient noise pulses. The state S_2 models the decaying oscillatory part of a noise pulse that usually follows the initial pulse of a transient noise. A codebook of waveforms in states S_1 and S_2 can model a variety of different noise pulses. Note that, in the HMM model of Figure 13.4, the self-loop transition provides a mechanism for the modelling of the variations in the duration of each noise pulse segment. The skip-state transitions provide a mechanism for the modelling of those noise pulses that do not exhibit either the initial nonlinear pulse or the decaying oscillatory part.

A hidden Markov model of noise can be employed for both the detection and the removal of transient noise pulses. As described in Section 13.3.3, the maximum-likelihood state-sequence of the noise HMM provides an estimate of the state of the noise at each time instant. The estimates of the states of the signal and the noise can be used for the implementation of an optimal state-dependent signal restoration algorithm.

13.3 DETECTION OF NOISE PULSES

For the detection of a pulse process, $n(m)$, observed in an additive signal, $x(m)$, the signal and the pulse can be modelled as

$$y(m) = b(m)n(m) + x(m) \quad (13.8)$$

where $b(m)$ is a binary ‘indicator’ process that signals the presence or absence of a noise pulse. Using the model of Equation (13.8), the detection of a noise pulse process can be considered as the estimation of the underlying binary-state noise-indicator process, $b(m)$. In this section, we consider three different methods for detection of transient noise pulses, using the noise template model within a matched filter, the linear predictive model of noise, and the hidden Markov model described in Section 13.2.

13.3.1 MATCHED FILTER FOR NOISE PULSE DETECTION

The inner product of two signal vectors provides a measure of the similarity of the signals. Since filtering is basically an inner product operation, it follows that the output of a filter should provide a measure of similarity of the filter input and the filter impulse response. The classical method for detection of a signal is to use a filter whose impulse response is *matched* to the shape of the signal to be detected. The derivation of a matched filter for the detection of a pulse, $n(m)$, is based on maximisation of the amplitude of the filter output when the input contains the pulse $n(m)$. The matched filter for the detection of a pulse, $n(m)$, observed in a ‘background’ signal, $x(m)$, is defined as

$$H(f) = K \frac{N^*(f)}{P_{xx}(f)} \quad (13.9)$$

where $P_{xx}(f)$ is the power spectrum of $x(m)$ and $N^*(f)$ is the complex conjugate of the spectrum of the noise pulse. When the ‘background’ signal process, $x(m)$, is a zero mean uncorrelated signal with variance σ_x^2 , the matched filter for detection of the transient noise pulse, $n(m)$, becomes

$$H(f) = \frac{K}{\sigma_x^2} N^*(f) \quad (13.10)$$

The impulse response of the matched filter corresponding to Equation (13.10) is given by

$$h(m) = Cn(-m) \quad (13.11)$$

where the scaling factor C is given by $C = K/\sigma_x^2$. Let $z(m)$ denote the output of the matched filter. In response to an input noise pulse, the filter output is given by the convolution relation

$$z(m) = Cn(-m) * n(m) \quad (13.12)$$

where the asterisk $*$ denotes convolution. In the frequency domain Equation (13.12) becomes

$$Z(f) = N(f)H(f) = C |N(f)|^2 \quad (13.13)$$

The matched filter output $z(m)$ is passed through a nonlinearity and a decision is made on the presence or the absence of a noise pulse as

$$\hat{b}(m) = \begin{cases} 1 & \text{if } |z(m)| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \quad (13.14)$$

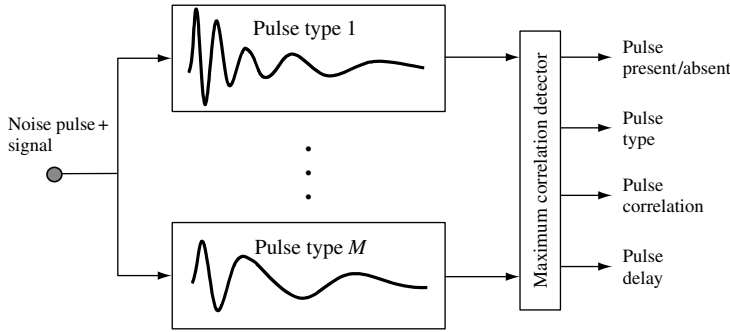


Figure 13.5 A bank of matched filters for detection of transient noise pulses.

In Equation (13.14), when the matched filter output exceeds a threshold, the detector flags the presence of the signal at the input. Figure 13.5 shows a noise pulse detector composed of a bank of M different matched filters. The detector signals the presence or the absence of a noise pulse. If a pulse is present then additional information provides the type of the pulse, the maximum cross-correlation of the input and the noise pulse template, and a time delay that can be used to align the input noise and the noise template. This information can be used for subtraction of the noise pulse from the noisy signal, as described in Section 13.4.1.

13.3.2 NOISE DETECTION BASED ON INVERSE FILTERING

The initial part of a transient noise pulse is often a relatively short and sharp impulsive-type event, which can be used as a distinctive feature for the detection of the noise pulses. The detectability of a sharp noise pulse, $n(m)$, observed in a correlated ‘background’ signal, $y(m)$, can often be improved by using a differencing operation, which has the effect of enhancing the relative amplitude of the impulsive-type noise. The differencing operation can be accomplished by an inverse linear predictor model of the background signal, $y(m)$. An alternative interpretation is that the inverse filtering is equivalent to a spectral whitening operation: it affects the energy of the signal spectrum, whereas the theoretically flat spectrum of the impulsive noise is largely unaffected. The use of an inverse linear predictor for the detection of an impulsive-type event was considered in detail in Section 12.4. Note that the inverse filtering operation reduces the detection problem to that of detecting a pulse in additive white noise.

13.3.3 NOISE DETECTION BASED ON HMM

In the three-state hidden Markov model of a transient noise pulse process described in Section 13.2.3, the states S_0 , S_1 and S_2 correspond to the noise-absent state, the initial noise pulse state and the decaying oscillatory noise state, respectively. As described in Chapter 5, an HMM, denoted by \mathcal{M} , is defined by a set of Markovian state transition probabilities and Gaussian state observation pdfs. The statistical parameters of the HMM of a noise pulse process can be obtained from a sufficiently large number of training examples of the process.

Given an observation vector, $\mathbf{y} = [y(0), y(1), \dots, y(N - 1)]$, the maximum likelihood state sequence $\mathbf{s} = [s(0), s(1), \dots, s(N - 1)]$, of the HMM \mathcal{M} is obtained as

$$s_{\text{ML}} = \arg \max_s f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\mathbf{s}, \mathcal{M}) \tag{13.15}$$

where, for a hidden Markov model, the likelihood of an observation sequence, $f_{\mathbf{Y}|\mathbf{S}}(\mathbf{y}|\mathbf{s}, \lambda)$, can be expressed as

$$\begin{aligned} & f_{\mathbf{Y}|\mathbf{S}} [y(0), y(1), \dots, y(N - 1)|s(0), s(1), \dots, s(N - 1)] \\ &= \pi_{s(0)} f_{s(0)} [y(0)] \cdot a_{s(0),s(1)} f_{s(1)} [y(1)] \cdots a_{s(N-2),s(N-1)} f_{s(N-1)} [y(N - 1)] \end{aligned} \tag{13.16}$$

where $\pi_{s(i)}$ is the initial state probability, $a_{s(i),s(j)}$ is the probability of a transition from state $s(i)$ to state $s(j)$, and $f_{s(i)} [y(i)]$ is the state observation pdf for the state $s(i)$. The maximum-likelihood state sequence, s_{ML} , derived using the Viterbi algorithm, is an estimate of the underlying states of the noise pulse process, and can be used as a detector of the presence or absence of a noise pulse.

13.4 REMOVAL OF NOISE PULSE DISTORTIONS

In this section, we consider two methods for the removal of transient noise pulses: (a) an adaptive noise subtraction method; and (b) an autoregressive model-based restoration method. The noise removal methods assume that a detector signals the presence or the absence of a noise pulse, and provides additional information on the timing and the underlying states of the noise pulse.

13.4.1 ADAPTIVE SUBTRACTION OF NOISE PULSES

The transient noise removal system shown in Figure 13.6 is composed of a matched filter for detection of noise pulses, a linear adaptive noise subtractor for cancellation of the linear transitory part of a noise pulse, and an interpolator for the replacement of samples irrevocably

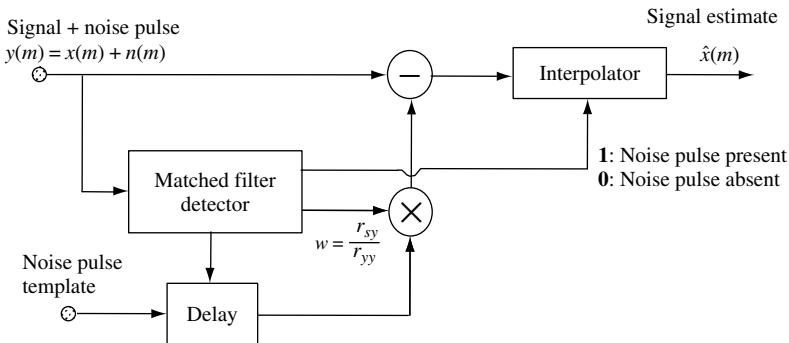


Figure 13.6 Transient noise pulse removal system.

distorted by the initial part of each pulse. Let $x(m)$, $n(m)$ and $y(m)$ denote the signal, the noise pulse and the noisy signal, respectively; the noisy signal model is

$$y(m) = x(m) + b(m) n(m) \quad (13.17)$$

where the binary indicator sequence, $b(m)$, indicates the presence or the absence of a noise pulse. Assume that each noise pulse, $n(m)$, can be modelled as the amplitude-scaled and time-shifted version of the noise pulse template $\bar{n}(m)$, so that

$$n(m) \approx w\bar{n}(m - D) \quad (13.18)$$

where w is an amplitude scalar and the integer D denotes the relative delay (time shift) between the noise pulse template and the detected noise. From Equations (13.17) and (13.18) the noisy signal can be modelled:

$$y(m) \approx x(m) + w\bar{n}(m - D) \quad (13.19)$$

From Equation (13.19) an estimate of the signal, $x(m)$, can be obtained by subtracting an estimate of the noise pulse from that of the noisy signal:

$$\hat{x}(m) = y(m) - w\bar{n}(m - D) \quad (13.20)$$

where the time delay D required for time-alignment of the noisy signal, $y(m)$, and the noise template, $\bar{n}(m)$, is obtained from the cross-correlation function *CCF* as

$$D = \arg \max_k \{CCF[y(m), \bar{n}(m - k)]\} \quad (13.21)$$

When a noise pulse is detected, the time lag corresponding to the maximum of the cross-correlation function is used to delay and time-align the noise pulse template with the noise pulse. The template energy is adaptively matched to that of the noise pulse by an adaptive scaling coefficient, w . The scaled and time-aligned noise template is subtracted from the noisy signal to remove linear additive distortions. The adaptive scaling coefficient, w , is estimated as follows. The correlation of the noisy signal $y(m)$ with the delayed noise pulse template $\bar{n}(m - D)$ gives

$$\begin{aligned} \sum_{m=0}^{N-1} y(m)\bar{n}(m - D) &= \sum_{m=0}^{N-1} [x(m) + w\bar{n}(m - D)]\bar{n}(m - D) \\ &= \sum_{m=0}^{N-1} x(m)\bar{n}(m - D) + w \sum_{m=0}^{N-1} \bar{n}(m - D)\bar{n}(m - D) \end{aligned} \quad (13.22)$$

where N is the pulse template length. Since the signal $x(m)$ and the noise $n(m)$ are uncorrelated, the term $\sum x(m)\bar{n}(m - D)$ on the right-hand side of Equation (13.22) is small, and we have

$$w \approx \frac{\sum_m y(m)\bar{n}(m - D)}{\sum_m \bar{n}^2(m - D)} \quad (13.23)$$

Note that, when a false detection of a noise pulse occurs, the cross-correlation term and hence the adaptation coefficient, w , could be small. This will keep the signal distortion resulting from false detections to a minimum.

Samples that are irrevocably distorted by the initial scratch pulse are discarded and replaced by one of the signal interpolators introduced in Chapter 10. When there is no noise pulse, the coefficient w is zero, the interpolator is bypassed and the input signal is passed through unmodified. Figure 13.7(b) shows the result of processing the noisy signal of Figure 13.7(a). The linear oscillatory noise is completely removed by the adaptive subtraction method. For this signal 80 samples irrevocably distorted by the initial scratch pulse were discarded and interpolated.

13.4.2 AR-BASED RESTORATION OF SIGNALS DISTORTED BY NOISE PULSES

A model-based approach to noise detection/removal provides a more compact method for characterisation of transient noise pulses, and has the advantage that closely spaced pulses can be modelled as the response of the model to a number of closely spaced input impulses. The signal $x(m)$ is modelled as the output of an AR model of order P_1 as

$$x(m) = \sum_{k=1}^{P_1} a_k x(m-k) + e(m) \quad (13.24)$$

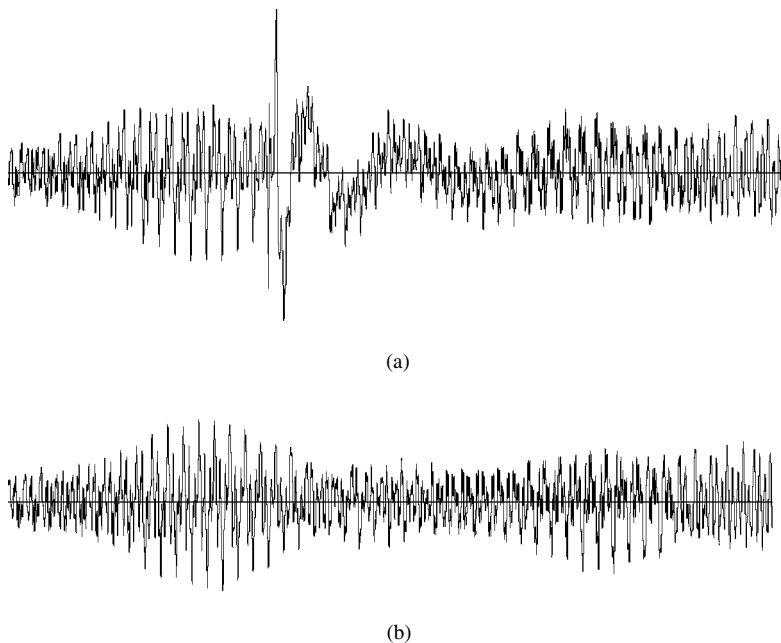


Figure 13.7 (a) A signal from an old gramophone record with a scratch noise pulse. (b) The restored signal.

Assuming that $e(m)$ is a zero-mean uncorrelated Gaussian process with variance σ_e^2 , the pdf of a vector, \mathbf{x} , of N successive signal samples of an autoregressive process with parameter vector, \mathbf{a} , is given by

$$f_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}\right) \quad (13.25)$$

where the elements of the matrix \mathbf{A} are composed of the coefficients a_k of the linear predictor model, as described in Section 8.4. In Equation (13.25), it is assumed that the P_1 initial samples are known. The AR model for a single noise pulse waveform $n(m)$ can be written as

$$n(m) = \sum_{k=1}^{P_2} c_k n(m-k) + A\delta(m) \quad (13.26)$$

where c_k is the model coefficient, P_2 is the model order, and the excitation is assumed to be an impulse of amplitude A . A number of closely spaced and overlapping noise pulses can be modelled as

$$n(m) = \sum_{k=1}^{P_2} a_k n(m-k) + \sum_j^M A_j \delta(m-T_j) \quad (13.27)$$

where it is assumed that T_k is the start of the k th excitation pulse in a burst of M pulses. A linear predictor model proposed by Godsill is driven by a binary-state excitation. The excitation waveform has two states: in state '0', the excitation is a zero-mean Gaussian process of variance σ_0^2 , and in state '1', the excitation is a zero-mean Gaussian process of variance $\sigma_1^2 \gg \sigma_0^2$. In state '1', the model generates a short-duration large amplitude excitation that largely models the transient pulse. In state '0', the model generates a low excitation that partially models the inaccuracies of approximating a nonlinear system by an AR model. The composite excitation signal can be written as

$$e_n(m) = [b(m)\sigma_1 + \bar{b}(m)\sigma_0]u(m) \quad (13.28)$$

where $u(m)$ is an uncorrelated zero-mean Gaussian process of unit variance, $b(m)$ is a binary sequence that indicates the state of the excitation, and $\bar{b}(m)$ is the binary complement of $b(m)$. When $b(m) = 1$ the excitation variance is σ_1^2 and when $b(m) = 0$, the excitation variance is σ_0^2 . The binary-state variance of $e_n(m)$ can be expressed as

$$\sigma_{e_n}^2(m) = b(m)\sigma_1^2 + \bar{b}(m)\sigma_0^2 \quad (13.29)$$

Assuming that the excitation pattern $\mathbf{b} = [b(m)]$ is given, the pdf of an N sample noise pulse \mathbf{x} is

$$f_{\mathbf{N}}(\mathbf{n}|\mathbf{b}) = \frac{1}{(2\pi)^{N/2} |\Lambda_{e_n e_n}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{n}^T \mathbf{C}^T \Lambda_{e_n e_n}^{-1} \mathbf{C} \mathbf{n}\right) \quad (13.30)$$

where the elements of the matrix \mathbf{C} are composed of the coefficients, c_k , of the linear predictor model, as described in Section 8.4. The posterior pdf of the signal \mathbf{x} given the noisy observation \mathbf{y} , $f_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$, can be expressed, using Bayes's rule, as

$$\begin{aligned} f_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) &= \frac{1}{f_{\mathbf{Y}}(\mathbf{y})} f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) \\ &= \frac{1}{f_{\mathbf{Y}}(\mathbf{y})} f_{\mathbf{N}}(\mathbf{y} - \mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) \end{aligned} \quad (13.31)$$

For a given observation, $f_{\mathbf{Y}}(\mathbf{y})$ is a constant. Substitution of Equations (13.30) and (13.25) in Equation (13.31) yields

$$\begin{aligned} f_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) &= \frac{1}{f_{\mathbf{Y}}(\mathbf{y})} \frac{1}{(2\pi\sigma_c)^N |\Lambda_{e_n e_n}|^{1/2}} \\ &\times \exp \left[-\frac{1}{2} (\mathbf{y} - \mathbf{x})^T \mathbf{C}^T \Lambda_{e_n e_n}^{-1} \mathbf{C} (\mathbf{y} - \mathbf{x}) - \frac{1}{2\sigma_c^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} \right] \end{aligned} \quad (13.32)$$

The MAP solution obtained by maximisation of the log posterior function with respect to the undistorted signal \mathbf{x} is given by

$$\hat{\mathbf{x}}^{\text{MAP}} = (\mathbf{A}^T \mathbf{A} / \sigma_c^2 + \mathbf{C}^T \Lambda_{e_n e_n}^{-1} \mathbf{C})^{-1} \mathbf{C}^T \Lambda_{e_n e_n}^{-1} \mathbf{C} \mathbf{y} \quad (13.33)$$

where $\hat{\mathbf{x}}^{\text{MAP}}$ is the MAP interpolation.

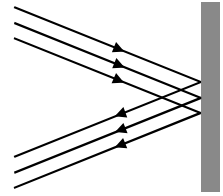
13.5 SUMMARY

In this chapter, we considered the modelling, detection and removal of transient noise pulses. Transient noise pulses are nonstationary events similar to impulsive noise, but usually occur less frequently and have a longer duration than impulsive noise. An important observation in the modelling of transient noise is that the noise can be regarded as the impulse response of a communication channel, and hence may be modelled by one of a number of statistical methods used in the of modelling communication channels. In Section 13.2, we considered several transient noise pulse models including a template-based method, an AR model-based method and a hidden Markov model. In Sections 13.2 and 13.3, these models were applied to the detection and removal of noise pulses.

BIBLIOGRAPHY

- Godsill S.J. (1993), The restoration of degraded audio signals. Ph.D. thesis, Cambridge University, Cambridge.
 Vaseghi S.V. (1987), Algorithm for restoration of archived gramophone recordings. Ph.D. thesis, Cambridge University, Cambridge.

14



Echo Cancellation

14.1 Introduction: Acoustic and Hybrid Echoes	352
14.2 Telephone Line Hybrid Echo	353
14.3 Hybrid Echo Suppression	355
14.4 Adaptive Echo Cancellation	356
14.5 Acoustic Echo	360
14.6 Sub-band Acoustic Echo Cancellation	363
14.7 Multiple-input Multiple-output Echo Cancellation	365
14.8 Summary	368
Bibliography	368

Echo is the repetition of a waveform due to reflections from points where the characteristics of the medium through which the wave propagates change. In telecommunications, echo can degrade the quality of service, and echo cancellation is an essential part of communications systems. The development of echo reduction began in the late 1950s, and continues today as new integration of landline, wireless cellular networks, internet, multi-input multi-output communication devices and teleconferencing systems place additional requirements on the performance of echo cancellers.

There are two types of echo in communication systems: acoustic echo and telephone line echo. Acoustic echo results from a feedback path set up between the loudspeaker and the microphone in a mobile phone, hands-free phone, teleconference or hearing aid system. Acoustic echo may be reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through multiple paths.

Telephone line echoes result from an impedance mismatch at the telephone exchange hybrids where the subscriber's two-wire line is connected to a four-wire line. The perceptual effects of an echo depend on the time delay between the incident and reflected waves, the strength of the reflected waves and the number of paths through which the waves are reflected. Telephone line echoes, and acoustic feedback echoes in teleconference and hearing aid systems, are undesirable and annoying and can be severely disruptive. In this chapter we study some methods for removing line echo from telephone and data telecommunications systems, and acoustic feedback echoes from microphone–loudspeaker systems.

14.1 INTRODUCTION: ACOUSTIC AND HYBRID ECHOES

Echo is the reflection of a signal from the points or surfaces where the characteristics of the medium through which the signal propagates change. Echo is usefully employed for detection, exploration and navigation purposes in electronic detection and imaging instruments such as in sonar, ultrasonic imaging, infrared imaging and radar, and by some animals such as bats and dolphins.

In telecommunications, echo can severely affect the quality and intelligibility of voice conversation in telephone, teleconference or cabin communication systems. The perceived effect of an echo depends on its amplitude and time delay. In general, echoes with appreciable amplitudes and a delay of more than 1 ms can be noticeable. Provided the round-trip delay of the echo signal is of the order of a few milliseconds, echo gives a telephone call a perceived sense of ‘liveliness’. However, echoes become increasingly annoying and objectionable with increasing round-trip delay and amplitude, in particular for delays of more than 20 ms. Above a delay of 200 ms, echoes can be disruptive for voice communication

Echo cancellation is an important aspect of the design of modern telecommunications systems such as conventional wire-line telephones, hands-free phones, cellular mobile (wireless) phones, teleconference systems and in-car cabin communication systems. There are two types of echo in a telephone system (Figure 14.1):

- (1) acoustic echo due to acoustic coupling between the speaker and the microphone in hands-free phones, mobile phones and teleconference systems;
- (2) electrical line echo due to mismatch at the hybrid circuit connecting a two-wire subscriber line to a four-wire trunk line in the public switched telephone network.

In telephone landlines echoes are mostly due to the impedance mismatch at the point of connection of the two-wire local line to the four-wire trunk line. In the early days of expansion of telephone networks, the cost of running a four-wire line from the local exchange to subscribers’ premises was considered uneconomical. Hence, at the exchange the four-wire trunk lines are converted to two-wire subscribers’ local lines using a two/four-wire hybrid bridge circuit. At the receiver, owing to any imbalance between the four/two-wire bridge circuit, some of the signal energy of the four-wire circuit is bounced back towards the transmitter, constituting an echo signal. If the echo is more than a few milliseconds long then it becomes noticeable, and can be annoying and disruptive.

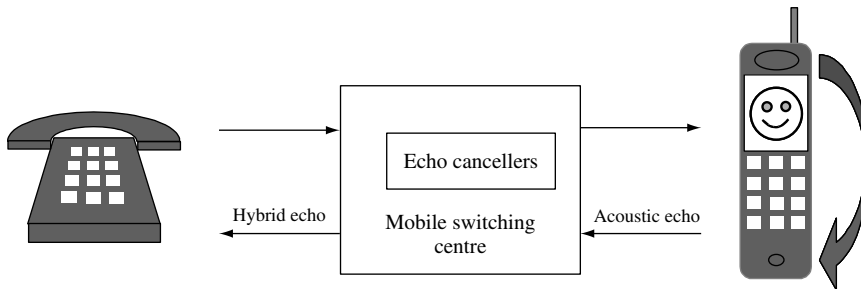


Figure 14.1 Illustration of sources of echo in a mobile-to-landline system.

In digital mobile phone systems, the echo is often due to the acoustic feedback coupling between the speaker and the microphone on the handset. In mobile phones the voice signals are processed at two points in the network: first, at the voice coder on the hand sets, the speech signals are digitised, divided into frames, compressed, coded and modulated; then the signal is processed at the radio frequency interface of the network. The total delay introduced by the various stages of digital signal processing range from 80 to 100 ms, resulting in a total round-trip delay of 160–200 ms for any echo. A delay of this magnitude will make any significant echo disruptive to the communication process. Owing to the inherent processing delay in digital mobile communication systems, it is essential and mandatory to employ echo cancellers in mobile phone switching centres.

14.2 TELEPHONE LINE HYBRID ECHO

Hybrid echo is the main source of echo generated from the public-switched telephone network (PSTN). Echoes on a telephone line are due to the reflection of speech signals at the points of impedance mismatch on the connecting circuits. Conventionally, telephones in a given geographical area are connected to an exchange by a two-wire twisted line, called the subscriber's line, which serves to receive and transmit signals. Both transmit and receive signals are present on the two-wire lines of the subscriber loop. In a conventional system a local call is set up by establishing a direct connection, at the telephone exchange, between two subscribers' loops. For a local call, there is usually no noticeable echo, either because there is no significant impedance mismatch on the connecting two-wire local lines or because the distances are relatively small and the resulting low-delay echoes (less than 30 ms) are perceived as a slight amplification and 'livening' effect. For long-distance communication between two exchanges, it is necessary to use repeaters to amplify the speech signals; therefore a separate two-wire telephone line is required for each direction of transmission.

To establish a long-distance call, at each end, a two-wire subscriber's line must be connected to a four-wire line at the exchange, as illustrated in Figure 14.2. The device that connects the two-wire subscriber's loop to the four-wire line is called a hybrid, and is shown in Figure 14.3. As shown, the hybrid is basically a three-port bridge circuit that separates the transmit and receive signals into two separate pairs of wires. If the hybrid bridge circuit were perfectly balanced, then there would be no reflection of signal or echo. However, each

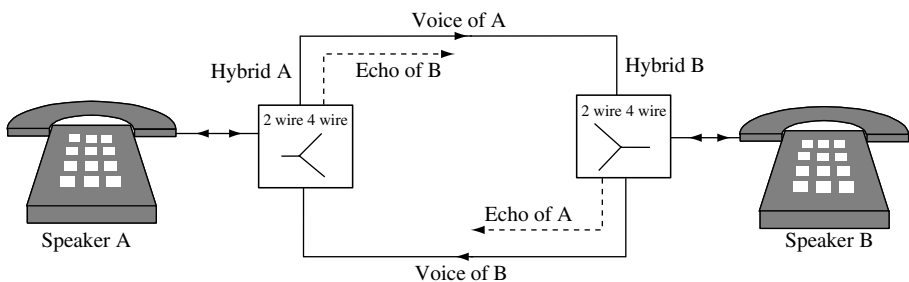


Figure 14.2 Illustration of a telephone call set up by connection of two-wire subscriber's phone lines via hybrids to four-wire lines at the exchange.

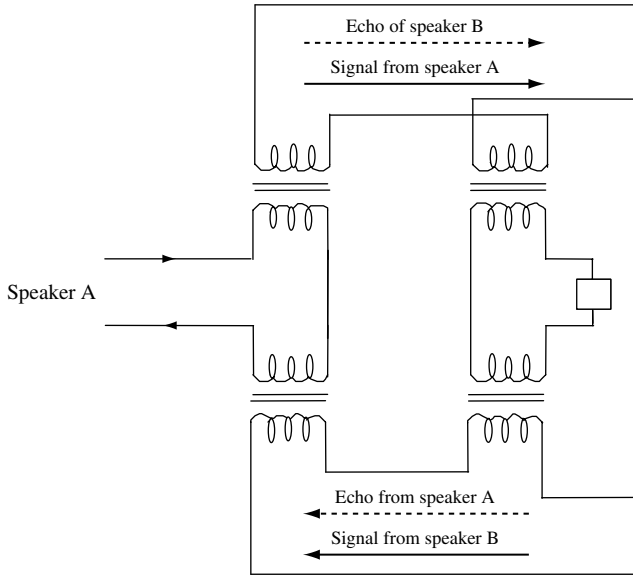


Figure 14.3 A two-wire to four-wire hybrid connection circuit.

hybrid circuit serves a number of subscribers' lines. The subscribers' lines do not all have the same length and impedance characteristics; therefore it is not possible to achieve perfect balance for all subscribers at the hybrids. When the bridge is not perfectly balanced, some of the signal energy on the receiving four-wire lines becomes coupled back onto itself and produces an echo.

14.2.1 ECHO: THE SOURCES OF DELAY IN TELEPHONE NETWORKS

The delay of an echo is the round trip time taken for the signal to arrive back at the source. End-to-end propagation delay is the sum of delays required for the voice of one speaker to propagate through different network devices and network links to reach the listener. Round-trip delay of an echo is the propagation time from the transmitter to the receiver and then back to transmitter.

For example when a long-distance call is made via a mobile phone and a satellite, the round-trip echo delay can be as long as, or sometimes even more than, 600 ms, and echoes can become disruptive. For this reason the employment of echo cancellers in mobile switching centres and satellite networks is mandatory.

The sources of delay in PSTN telephone network are as follows:

- speech coder delay, 2.5–40 ms;
- inter-process hand-offs delay, about 10 ms at each end;
- transmission line delay, about 1 ms per 100 miles of cable;
- satellite links delay, 250–300 ms – multiple hops can yield longer delays;

- packet delay, from 0.5 ms for a 128 K packet on a 2 Mbps line, to 128.6 ms for a 1024 K packet on a 64 kbps line;
- voice-over IP (VoIP) gateway node, 50–100 ms;
- decompression delay, typically 10 ms or less.

Acoustic echo can be longer than line echo. The duration of acoustic echo depends on the dimensions of the room and the number of reflections off the walls that the echo goes through. For example, sound travels at a speed of 340 m/s at a room temperature of 25°C. Hence the time taken for sound to travel 1 m will be about $1/340 \approx 2.94$ ms. A trip of 10 m from speaker to microphone will take about 29.4 ms and then to this must be added the delay for coding and transmission through the communication network as described above.

14.2.2 ECHO RETURN LOSS

The intensity of echo is measured in terms the echo return loss (ERL), defined as the power ratio, in dB, of the transmitted signal to that of the returned echo as

$$\text{ERL} = 10 \log_{10} \left(\frac{\text{transmitted signal power}}{\text{echo return signal power}} \right) \quad (\text{dB}) \quad (14.1)$$

The higher the echo return loss is, the lower will be the power of the echo. The echo return loss enhancement (ERLE) is the difference in ERL before and after application of the echo cancellation.

14.3 HYBRID ECHO SUPPRESSION

The development of echo reduction began in the late 1950s with the advent of echo suppression systems. Echo suppressors were first employed to manage the echo generated primarily in satellite circuits. An echo suppressor (Figure 14.4) is primarily a switch that lets the speech signal through during the speech-active periods and attenuates the line echo

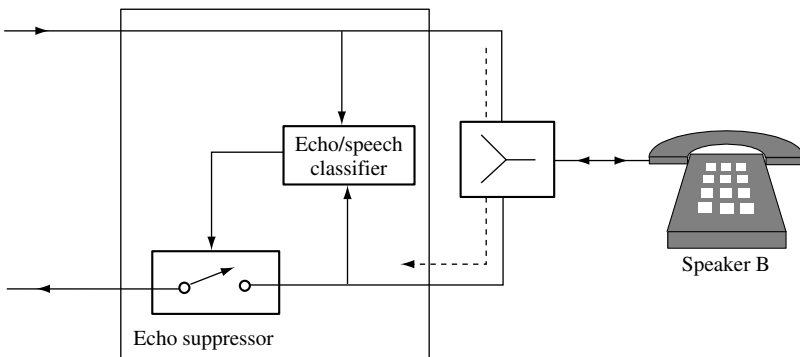


Figure 14.4 Block diagram illustration of an echo suppression system.

during the speech-inactive periods. A line echo suppresser is controlled by a speech/echo detection device. The echo detector monitors the signal levels on the incoming and outgoing lines, and decides if the signal on a line from, say, speaker B to speaker A is the speech from the speaker B to the speaker A, or the echo of speaker A. If the echo detector decides that the signal is an echo then the signal is heavily attenuated. There is a similar echo suppression unit from speaker A to speaker B.

The performance of an echo suppresser depends on the accuracy of the echo/speech classification subsystem. Echo of speech often has a smaller amplitude than the speech signal, but otherwise it has mainly the same spectral characteristics and statistics as those of the speech. Therefore the only basis for discrimination of speech from echo is the signal level. As a result, the speech/echo classifier may wrongly classify and let through high-level echoes as speech, or attenuate low-level speech as echo. For terrestrial circuits, echo suppressers have been well designed, with an acceptable level of false decisions and a good performance. The performance of an echo suppresser depends on the time delay of the echo. In general, echo suppressers perform well when the round-trip delay of the echo is less than 100 ms. For a conversation routed via a geostationary satellite, the round-trip delay may be as much as 600 ms. Such long delays can change the pattern of conversation and result in a significant increase in speech/echo classification errors. When the delay is long, echo suppressers fail to perform satisfactorily, and this results in choppy first syllables and artificial volume adjustment. A system that is effective with both short and long time delays is the *adaptive echo canceller*, introduced next.

14.4 ADAPTIVE ECHO CANCELLATION

Echo cancellation was developed in the early 1960s by AT&T Bell Laboratories and later by COMSAT TeleSystems. The first echo cancellation systems were experimentally implemented across satellite communication networks to demonstrate network performance for long-distance calls.

Figure 14.5 illustrates the operation of an adaptive line echo canceller. The speech signal on the line from speaker A to speaker B is input to the four/two-wire hybrid B and to the echo canceller. The echo canceller monitors the signal on line from B to A and attempts to model the echo path and synthesise a replica of the echo of speaker A. This replica is used to

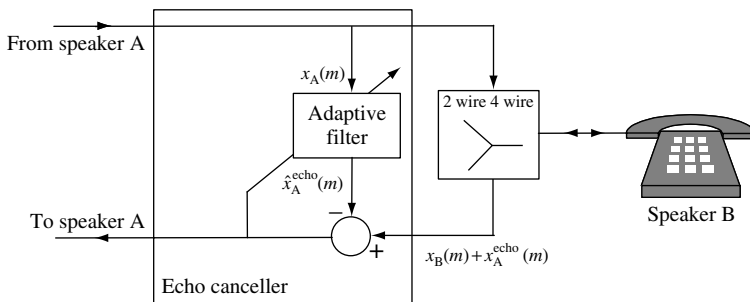


Figure 14.5 Block diagram illustration of an adaptive echo cancellation system.

subtract and cancel out the echo of speaker A on the line from B to A. The echo canceller is basically an adaptive linear filter. The coefficients of the filter are adapted so that the energy of the signal on the line is minimised. The echo canceller can be an infinite impulse response or a finite impulse response filter. The main advantage of an IIR filter is that a long-delay echo can be synthesised by a relatively small number of filter coefficients. In practice, echo cancellers are based on FIR filters. This is mainly due to the practical difficulties associated with the adaptation and stable operation of adaptive IIR filters.

Assuming that the signal on the line from speaker B to speaker A, $y_B(m)$, is composed of the speech of speaker B, $x_B(m)$, plus the echo of speaker A, $x_A^{\text{echo}}(m)$, we have

$$y_B(m) = x_B(m) + x_A^{\text{echo}}(m) \quad (14.2)$$

In practice, speech and echo signals are not simultaneously present on a phone line unless both speakers are speaking simultaneously. This, as pointed out below, can be used to simplify the adaptation process. Assuming that the truncated impulse response of the echo path is modelled by an FIR filter, the output estimate of the synthesised echo signal can be expressed as

$$\hat{x}_A^{\text{echo}}(m) = \sum_{k=0}^P h_k(m)x_A(m-k) \quad (14.3)$$

where $h_k(m)$ are the time-varying coefficients of an adaptive FIR filter model of the echo path and $\hat{x}_A^{\text{echo}}(m)$ is an estimate of the echo of speaker A on the line from speaker B to speaker A. The residual echo signal, or the error signal, after echo subtraction is given by

$$\begin{aligned} e(m) &= y_B(m) - \hat{x}_A^{\text{echo}}(m) \\ &= x_B(m) + x_A^{\text{echo}}(m) - \sum_{k=0}^P h_k(m)x_A(m-k) \end{aligned} \quad (14.4)$$

For those time instants when speaker A is talking and speaker B is listening and silent, and only echo is present from line B to A, we have

$$\begin{aligned} e(m) &= \tilde{x}_A^{\text{echo}}(m) = x_A^{\text{echo}}(m) - \hat{x}_A^{\text{echo}}(m) \\ &= x_A^{\text{echo}}(m) - \sum_{k=0}^P h_k(m)x_A(m-k) \end{aligned} \quad (14.5)$$

where $\tilde{x}_A^{\text{echo}}(m)$ is the residual echo. An echo canceller using an adaptive FIR filter is illustrated in Figure 14.6. The magnitude of the residual echo depends on the ability of the echo canceller to synthesise a replica of the echo, and this in turn depends on the adaptation algorithm discussed next.

14.4.1 ECHO CANCELLER ADAPTATION METHODS

The echo canceller coefficients, $h_k(m)$, are adapted to minimise the energy of the echo signal on a telephone line, say from speaker B to speaker A. Assuming that the speech

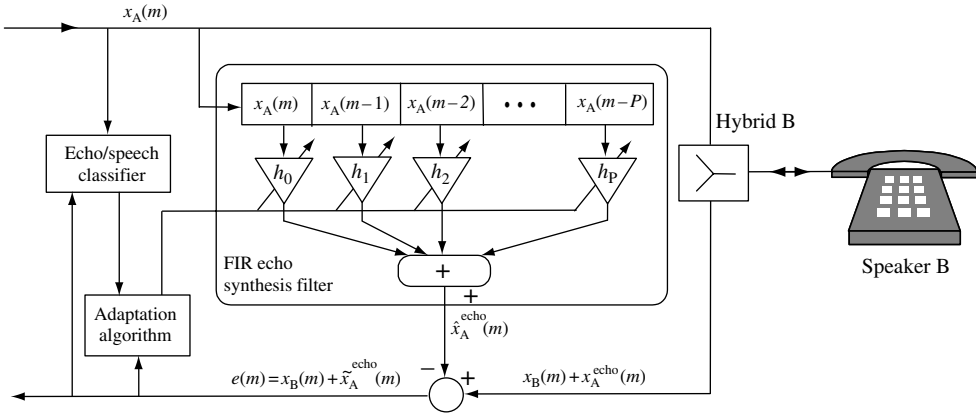


Figure 14.6 Illustration of an echo canceller using an adaptive FIR filter and incorporating an echo/speech classifier.

signals $x_A(m)$ and $x_B(m)$ are uncorrelated, the energy on the telephone line from B to A is minimised when the echo canceller output, $\hat{x}_A^{\text{echo}}(m)$, is equal to the echo, $x_A^{\text{echo}}(m)$, on the line. The echo canceller coefficients may be adapted using one of the variants of the recursive least square error or the least mean squared error adaptation methods. One of the most widely used algorithms for adaptation of the coefficients of an echo canceller is the normalised least mean square error (NLMS) method. The time-update equation describing the adaptation of the filter coefficient vector is

$$\mathbf{h}(m) = \mathbf{h}(m - 1) + \mu \frac{e(m)}{\mathbf{x}(m)_{\text{A}}^T \mathbf{x}_A(m)} \mathbf{x}_A(m) \tag{14.6}$$

where $\mathbf{x}_A(m) = [x_A(m), \dots, x_A(m - P)]$ and $\mathbf{h}(m) = [h_0(m), \dots, h_{P-1}(m)]$ are the input signal vector and the coefficient vector of the echo canceller, and $e(m)$ is the error signal that is the difference between the signal on the echo line and the output of the echo synthesiser. Note that the normalising quantity $\mathbf{x}(m)_{\text{A}}^T \mathbf{x}_A(m)$ is the energy of the input speech to the adaptive filter. The scalar μ is the adaptation step size, and controls the speed of convergence, the steady-state error and the stability of the adaptation process.

14.4.2 CONVERGENCE OF LINE ECHO CANCELLER

For satisfactory performance, the echo canceller should have a fast convergence rate, so that it can adequately track changes in the communication link and the signal characteristics. The convergence of an echo canceller is affected by the following factors:

- (1) *Nonstationary characteristics of telephone line and speech* – the echo characteristics depend on the impedance mismatch between the subscribers’ loop and the hybrids. Any changes in the connecting paths affect the echo characteristics and the convergence process. Also, as explained in Chapter 7, the nonstationary character and the eigenvalue

spread of the input speech signal of an LMS adaptive filter affect the convergence rates of the filter coefficients.

- (2) *Simultaneous conversations – double talks.* In a telephone conversation, usually the talkers do not speak simultaneously, and hence speech and echo are seldom present on a line at the same time. This observation simplifies the echo cancellation problem and substantially aids the correct functioning of adaptive echo cancellers. Problems arise during the periods when both speakers talk at the same time. This is because speech and its echo have similar characteristics and occupy basically the same bandwidth. When the reference signal contains both echo and speech, the adaptation process can lose track, and the echo cancellation process can attempt to cancel out and distort the speech signal. One method of avoiding this problem is to use a voice activity detector (VAD), and freeze the adaptation process during periods when speech and echo are simultaneously present on a line, as shown in Figure 14.6. In this system, the effect of a speech/echo misclassification is that the echo may not be optimally cancelled out. This is more acceptable than is the case in echo suppressors, where the effect of a misclassification is the suppression and loss of part of the speech.
- (3) *The adaptation algorithm* – most echo cancellers use variants of the LMS adaptation algorithm. The attractions of the LMS are its relatively low memory and computational requirements and its ease of implementation and monitoring. The main drawback of LMS is that it can be sensitive to the eigenvalue spread of the input signal and is not particularly fast in its convergence rate. However, in practice, LMS adaptation has produced effective line echo cancellation systems. The RLS error methods have a faster convergence rate and a better minimum mean square error performance. With the increasing availability of low-cost high-speed dedicated digital signal processing (DSP) processors, implementation of higher-performance and computationally intensive echo cancellers based on RLS are now feasible.

14.4.3 ECHO CANCELLATION FOR DIGITAL DATA TRANSMISSION

Echo cancellation becomes more complex with the increasing integration of wireline telephone systems and mobile cellular systems, and the use of digital transmission methods such as asynchronous transfer mode (ATM) for integrated transmission of data, image and voice. For example, in ATM based systems, the voice transmission delay varies depending on the route taken by the cells that carry the voice signals. This variable delay added to the delay inherent in digital voice coding complicates the echo cancellation process.

The 2-wire subscriber telephone lines that were originally intended to carry relatively low-bandwidth voice signals are now used to provide telephone users with high-speed digital data links and digital services such as video-on-demand and internet services using digital transmission methods such as the asynchronous digital subscriber line (ADSL). Traditionally, the bandwidth of the subscribers line is limited by low-pass filters at the core network to 3.4 kHz. Within this bandwidth, voice-band modems can provide data rates of around 30 kbps. However the copper wire itself has a much higher usable bandwidth extending into megahertz regions, although attenuation and interference increase with both the frequency and the length of the wire. Advanced signal processing and modulation scheme methods such as ADSL can achieve a 10 Mbps data rate over the 240 MHz bandwidth of subscriber's twisted-wire line.

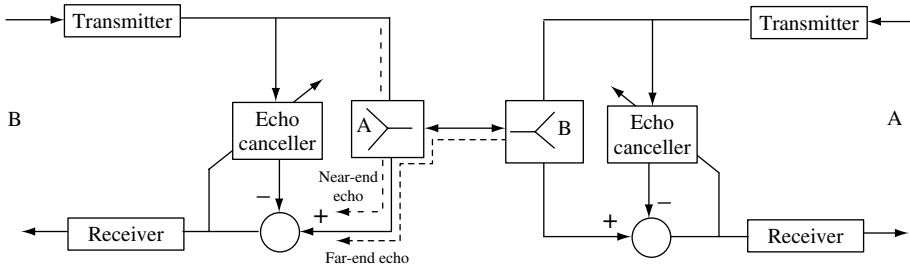


Figure 14.7 Echo cancellation in digital modems using 2-wire subscriber's loop.

Figure 14.7 shows a system for providing a full-duplex digital service over a two-wire subscriber's loop. To provide simultaneous transmission of data in both directions within the same bandwidth over the subscriber's line, echo cancellation is needed. The echoes on a line consist of the near-end echo, which loops back at the first or the near hybrid, and the far-end echo, which is the signal that loops back at a hybrid some distance away. The main purpose of the echo canceller is to cancel the near-end echo. Since the digital signal coming from a far end may be attenuated by 40–50 dB, the near echo on a high-speed data transmission line can be as much as 40–50 dB above the desired signal level. For reliable data communication the echo canceller must provide 50–60 dB attenuation of the echo signal so that the signal power remains at 10 dB above the echo.

14.5 ACOUSTIC ECHO

Acoustic echo results from a feedback path set up between the speaker and the microphone in a mobile phone, hands-free phone, teleconference or hearing aid system. Acoustic echo is reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through different paths. If the time delay is not too long, then the acoustic echo may be perceived as a soft reverberation, and may add to the artistic quality of the sound; concert halls and church halls with desirable reverberation characteristics can enhance the quality of a musical performance. However, acoustic echo is a well-known problem with hands-free telephones, teleconference systems, public address systems, mobile phones and hearing aids, and is due to acoustic feedback coupling of sound waves between the loudspeakers and microphones. Acoustic echo can result from a combination of direct acoustic coupling and multipath effect where the sound wave is reflected from various surfaces and then picked up by the microphone. In its worst case, acoustic feedback can result in howling if a significant proportion of the sound energy transmitted by the loudspeaker is received back at the microphone and circulated in the feedback loop. The overall round gain of an acoustic feedback loop depends on the frequency responses of the electrical and the acoustic signal paths. The undesirable effects of the electrical sections on the acoustic feedback can be reduced by designing systems that have a flat frequency response. The main problem is in the acoustic feedback path and the reverberating characteristics of the room. If the microphone–speaker–room system is excited at a frequency whose loop gain is greater than unity, then the signal is amplified each time it circulates round the loop, and feedback howling results. In practice, the howling is limited by the nonlinearity of the electronic system.

There are a number of methods for removing acoustic feedback. One method for alleviating the effects of acoustic feedback and room reverberations is to place a frequency shifter (or a phase shifter) in the electrical path of the feedback loop. Each time a signal travels round the feedback loop it is shifted by a few hertz before being re-transmitted by the loudspeaker. This method has some effect in reducing the howling, but it is not effective for removal of the overall echo of the acoustic feedback. Another approach is to reduce the feedback loop-gain at those frequencies where the acoustic feedback energy is concentrated. This may be achieved using adaptive notch filters to reduce the system gain at frequencies where acoustic oscillations occur. The drawback of this method is that, in addition to reducing the feedback, the notch filters also result in distortion of the desired signal frequencies.

The most effective method of acoustic feedback removal is the use of an adaptive feedback cancellation system. Figure 14.8 illustrates a model of an acoustic feedback environment, comprising a microphone, a loudspeaker and the reverberating space of a room. The z -transfer function of a linear model of the acoustic feedback environment may be expressed as

$$H(z) = \frac{G(z)}{1 - G(z)A(z)} \tag{14.7}$$

where $G(z)$ is the z -transfer function model for the microphone–loudspeaker system and $A(z)$ is the z -transfer function model of reverberations and multipath reflections of a room environment. Assuming that the microphone–loudspeaker combination has a flat frequency response with a gain G , Equation (14.7) can be simplified to

$$H(z) = \frac{G}{1 - GA(z)} \tag{14.8}$$

Note that in Equations (14.6)–(14.7), owing to the reverberating character of the room, the acoustic feedback path $A(z)$ is itself a feedback system. The reverberating characteristics of the acoustic environment may be modelled by an all-pole linear predictive model, or alternatively a relatively long FIR model.

The equivalent time-domain input/output relation for the linear filter model of Equation (14.8) is given by the following difference equation:

$$y(m) = \sum_{k=0}^P a_k(m)y(m - k) + Gx(m) \tag{14.9}$$

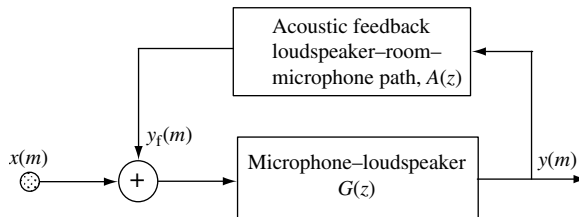


Figure 14.8 Configuration of a feedback model for a microphone–loudspeaker–room system.

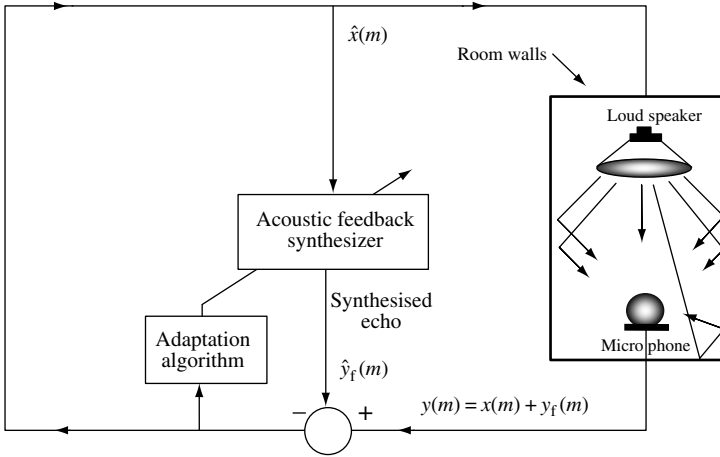


Figure 14.9 Illustration of adaptive acoustic feedback cancellation in a conference room environment.

where $a_k(m)$ is the coefficient of an all-pole linear feedback model of the reverberating room environment, G is the microphone–loudspeaker amplitude gain factor, and $x(m)$ and $y(m)$ are the time domain input and output signals of the microphone–loudspeaker system.

Figure 14.9 is an illustration of an acoustic feedback cancellation system. In an acoustic feedback environment, the total input signal to the microphone is given as the sum of any new input to the microphone $x(m)$ plus the unwanted acoustic feedback signal $y_f(m)$:

$$y(m) = x(m) + y_f(m) \quad (14.10)$$

The most successful acoustic feedback control systems are based on adaptive estimation and cancellation of the feedback signal. As in a line echo canceller, an adaptive acoustic feedback canceller attempts to synthesise a replica of the acoustic feedback at its output as

$$\hat{y}_f(m) = \sum_{k=0}^P a_k(m) y(m-k) \quad (14.11)$$

The filter coefficients are adapted to minimise the energy of an error signal, defined as

$$e(m) = x(m) + y_f(m) - \hat{y}_f(m) \quad (14.12)$$

The adaptation criterion is usually the minimum mean square error criterion and the adaptation algorithm is a variant of the LMS or the RLS method. The problem of acoustic echo cancellation is more complex than line echo cancellation for a number of reasons. First, acoustic echo is usually much longer (up to a second) than terrestrial telephone line echoes. In fact, the delay of an acoustic echo is similar to or more than a line echo routed via a geostationary satellite system.

The large delay of an acoustic echo path implies that impractically large filters on the order of a few thousand coefficients may be required. The stable and speedy adaptation of

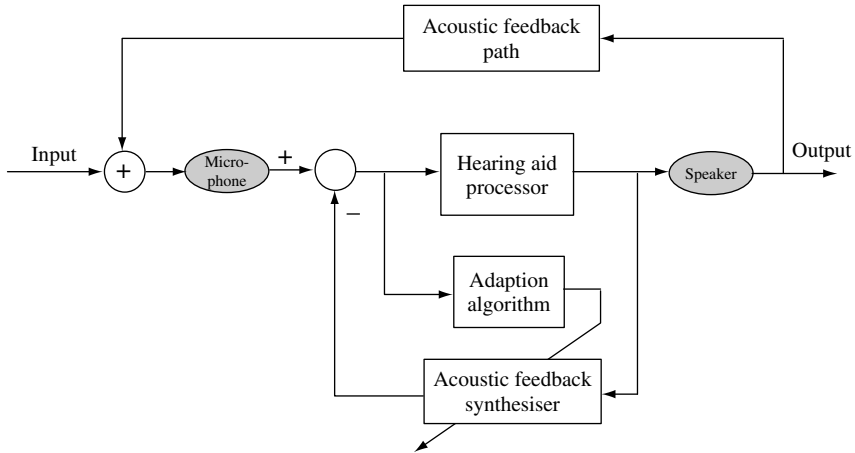


Figure 14.10 Configuration of an acoustic feedback canceller incorporated in a hearing aid system.

filters of such length presents a difficult problem. Secondly, the characteristics of an acoustic echo path is more nonstationary compared with that of a telephone line echo. For example, the opening or closing of a door, or people moving in or out of a room, can suddenly change the acoustic character of a conference room. Thirdly, acoustic echoes are due to signals reflected back from a multitude of different paths, off the walls, the floor, the ceiling, the windows, etc. Finally, the propagation and diffusion characteristics of the acoustic space of a room are nonlinear, and are not well approximated by a lumped FIR (or IIR) linear filter. In comparison, it is more reasonable to model the characteristics of a telephone line echo with a linear filter. In any case, for acoustic echo cancellation, the filter must have a large impulse response and should be able to quickly track fast changes in echo path characteristics.

An important application of acoustic feedback cancellation is in hearing aid systems. A hearing aid system can be modelled as a feedback system, as shown in Figure 14.10. The maximum usable gain of a hearing aid system is limited by the acoustic feedback between the microphone and the speaker. Figure 14.10 illustrates the configuration of a feedback canceller in a hearing aid system. The acoustic feedback synthesiser has the same input as the acoustic feedback path. An adaptation algorithm adjusts the coefficients of the synthesiser to cancel out the feedback signals picked up by the microphone, before the microphone output is fed into the speaker.

14.6 SUB-BAND ACOUSTIC ECHO CANCELLATION

In addition to the complex and varying nature of room acoustics, there are two main problems in acoustic echo cancellation. First, the echo delay is relatively long, and therefore the FIR echo synthesiser must have a large number of coefficients, say 2000 or more. Secondly, the long impulse response of the FIR filter and the large eigenvalue spread of the speech signals result in a slow, and uneven, rate of convergence of the adaptation process.

A sub-band-based echo canceller alleviates the problems associated with the required filter length and the speed of convergence. The sub-band-based system is shown in Figure 14.11.

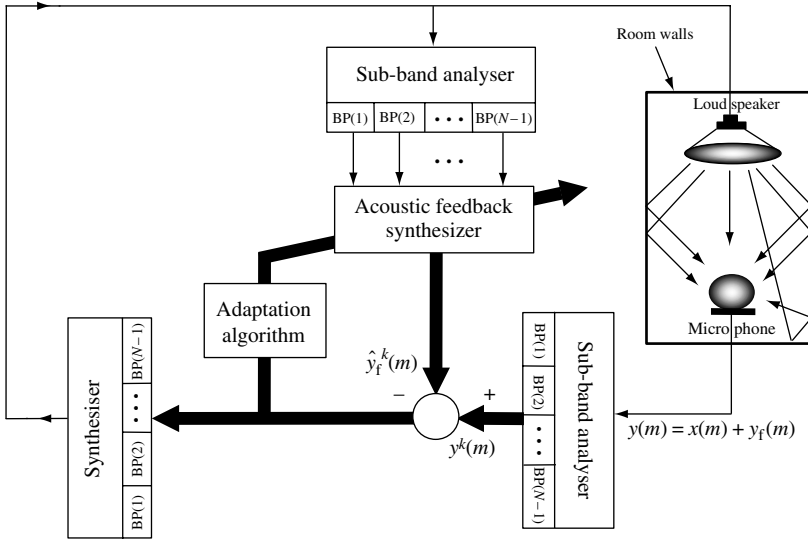


Figure 14.11 Configuration of a sub-band acoustic echo cancellation system.

The sub-band analyser splits the input signal into N sub-bands. Assuming that the sub-bands have equal bandwidth, each sub-band occupies only $1/N$ of the baseband frequency, and can therefore be decimated (down-sampled) without loss of information. For simplicity, assume that all sub-bands are down-sampled by the same factor, R . The main advantages of a sub-band echo canceller are a reduction in filter length and a gain in the speed of convergence as explained below:

- (1) *Reduction in filter length* – assuming that the impulse response of each sub-band filter has the same duration as the impulse response of the full-band FIR filter, the length of the FIR filter for each down-sampled sub-band is $1/R$ of the full band filter.
- (2) *Reduction in computational complexity* – the computational complexity of an LMS-type adaptive filter depends directly on the product of the filter length and the sampling rate. As for each sub-band, the number of samples per second and the filter length decrease with $1/R$, it follows that the computational complexity of each sub-band filter is $1/R^2$ of that of the full band filter. Hence the overall gain in computational complexity of a sub-band system is R^2/N of the full band system.
- (3) *Speed of convergence* – the speed of convergence depends on both the filter length and the eigenvalue spread of the signal. The speed of convergence increases with the decrease in the length of the FIR filter for each sub-band. A more important factor affecting the convergence of adaptive filter is the eigenvalue spread of the autocorrelation matrix of the input signal. As the spectrum of a signal becomes flatter, the spread of its eigenvalues decreases, and the speed of convergence of the adaptive filter increases. In general, the signal within each sub-band is expected to have a flatter spectrum than the full band signal. This aids the speed of convergence. However, it must be noted that the attenuation of sub-band filters at the edges of the spectrum of each band creates some very small eigenvalues.

14.7 MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO) ECHO CANCELLATION

Multiple-input multiple-output (MIMO) echo-cancellation systems have applications in car cabin communications systems, stereophonic teleconferencing systems and conference halls. Stereophonic echo cancellation systems have been developed relatively recently and MIMO systems are still the subject of ongoing research and development.

In a typical MIMO system there are P speakers and Q microphones in the room. As there is an acoustic feedback path set up between each speaker and each microphone, there are altogether $P \times Q$ such acoustic feedback paths that need to be modelled and estimated. The truncated impulse response of each acoustic path from loudspeaker i to microphone j is modelled by an FIR filter h_{ij} . The truncated impulse response of each acoustic path from a human speaker i to microphone j is modelled by an FIR filter, g_{ij} .

For a large number of speakers and microphones, the modelling and identification of the numerous acoustic channels becomes a major problem due to the correlations of the echo signals, from a common number of sources, propagating through different channels, as discussed below.

14.7.1 STEREOPHONIC ECHO CANCELLATION SYSTEMS

Figure 14.12 shows the configuration of the echo cancellation for a stereophonic communication system. There are two microphones and two loudspeakers at each end of the communication link. Each microphone receives the feedback echo from two different speakers through two different paths. In addition there are usually multipath reflections of

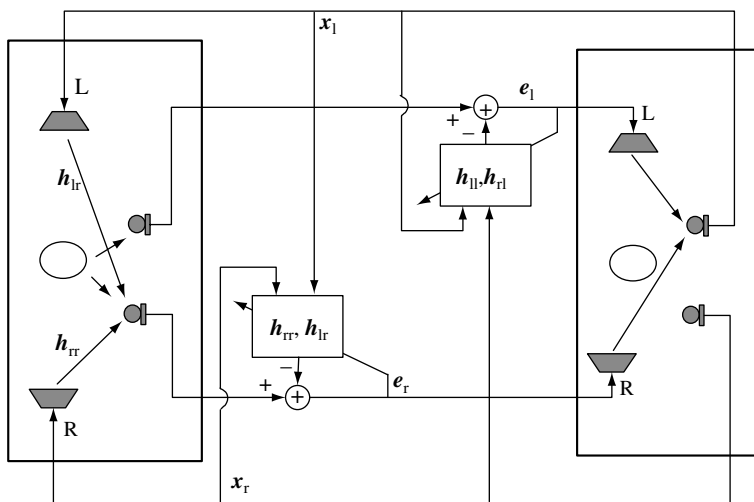


Figure 14.12 Illustration of the feedback signals and adaptive cancellation of acoustic feedbacks in a stereophonic echo-cancellation system.

sounds from walls. Let the speech signal $s(m)$ after reaching the right and left microphones be denoted as $x_r(m)$ and $x_l(m)$ respectively. We may write

$$x_r(m) = \mathbf{g}_r^T(m)s(m)$$

and

$$x_l(m) = \mathbf{g}_l^T(m)s(m) \quad (14.13)$$

where \mathbf{g}_r and \mathbf{g}_l are the truncated room impulse responses from the source speaker to the right and left microphones respectively.

In Figure 14.12, the truncated impulse responses of the acoustic feedback paths from the right and left loudspeakers to the right microphone are denoted as $\mathbf{h}_{rr}(m)$ and $\mathbf{h}_{lr}(m)$ respectively, and combined as $\mathbf{h}_r(m) = [\mathbf{h}_{rr}(m)\mathbf{h}_{lr}(m)]$. The signals from the right and left loudspeakers, $x_r(m)$ and $x_l(m)$, may be combined as $\mathbf{x}(m) = [x_r(m)x_l(m)]$. There exist similar paths from loudspeakers to each microphone which are not shown here in order to avoid overcrowding the figure. The synthesised replication of the echo signal in the right microphone is given by

$$\hat{x}_{\text{echo},r}(m) = \underbrace{\hat{\mathbf{h}}_{rr}^T(m-1)\mathbf{x}_r(m)}_{\substack{\text{Synthesised echo from right} \\ \text{loudspeaker to right microphone}}} + \underbrace{\hat{\mathbf{h}}_{lr}^T(m-1)\mathbf{x}_l(m)}_{\substack{\text{Synthesised echo from left} \\ \text{loudspeaker to right microphone}}} = \hat{\mathbf{h}}_r^T(m-1)\mathbf{x}(m) \quad (14.14)$$

The error signal composed of speech and the residue echo is given by

$$e_r(m) = y(m) - \hat{x}_{\text{echo},r}(m) \quad (14.15)$$

The NLMS adaptation method for FIR model of stereophonic echo paths is

$$\hat{\mathbf{h}}_r(m) = \hat{\mathbf{h}}_r(m-1) + \mu \frac{e(m)}{\mathbf{x}(m)^T \mathbf{x}(m)} \mathbf{x}(m) \quad (14.16)$$

Similar equations describing the echo and adaptive echo cancellation can be written for the left microphones.

14.7.1.1 Nonuniqueness Problem in MIMO Echo Channel Identification

A problem in MIMO echo cancellation systems is that the speech signals from different loudspeakers reaching a microphone are highly correlated. For the stereo echo cancellation, the loudspeakers' signals are $\mathbf{x}(m) = [x_r(m)x_l(m)]$ and the channels to be identified for example from right and left loudspeakers to say the right microphone are $\mathbf{h}_r(m) = [\mathbf{h}_{rr}(m)\mathbf{h}_{lr}(m)]$. The Wiener solution to this echo path estimation problem is given by

$$\hat{\mathbf{h}}_r = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xy} \quad (14.17)$$

where \mathbf{R}_{xx} is the autocorrelation matrix of $\mathbf{x}(m) = [x_r(m)x_l(m)]$. The problem is that due to the high correlation of $x_r(m)$ and $x_l(m)$ the autocorrelation matrix \mathbf{R}_{xx} is not of full rank matrix and hence the solution is not unique.

The nonuniqueness problem can also be explained in the frequency domain by considering the sum of the feedbacks from the loudspeakers into say the right microphone $X_{\text{echo},r}(f)$:

$$\begin{aligned} X_{\text{echo},r}(f) &= H_{lr}(f)G_l(f)S(f) + H_{rr}(f)G_r(f)S(f) \\ &= [H_{lr}(f)G_l(f) + H_{rr}(f)G_r(f)]S(f) \end{aligned} \tag{14.18}$$

Where $G_l(f)$, $G_r(f)$ are the frequency responses of the paths from the source to the transmitter microphones and $H_{lr}(f)$ and $H_{rr}(f)$ are the frequency responses of the loudspeakers feedback paths to the receiver’s right microphone. Note there are many combinations of different values of $G_l(f)$, $G_r(f)$, $H_{lr}(f)$ and $H_{rr}(f)$ that would satisfy equation (14.18). A solution to this problem is to decorrelate the stereo signals. The problem of stereophonic echo cancellation is the subject of ongoing research. A good paper on the problems of stereophonic echo cancellation is by Benesty J. (1998).

14.7.1.2 MIMO in Cabin Communication Systems

MIMO systems have application for in-car cabin communication systems (CCS) for multipurpose and large vehicles. The problems in CCS systems are background noise reduction and acoustic feedback cancellation. Figure 14.13 illustrates the block diagram configuration of a two-loudspeaker two-microphone system and the configurations of the FIR adaptive filters used for modelling the truncated impulse response of the acoustic feedback path from each loudspeaker via the cabin to each microphone. The synthesised feedback is subtracted from the signal received by each microphone in order to cancel feedback echo signals.

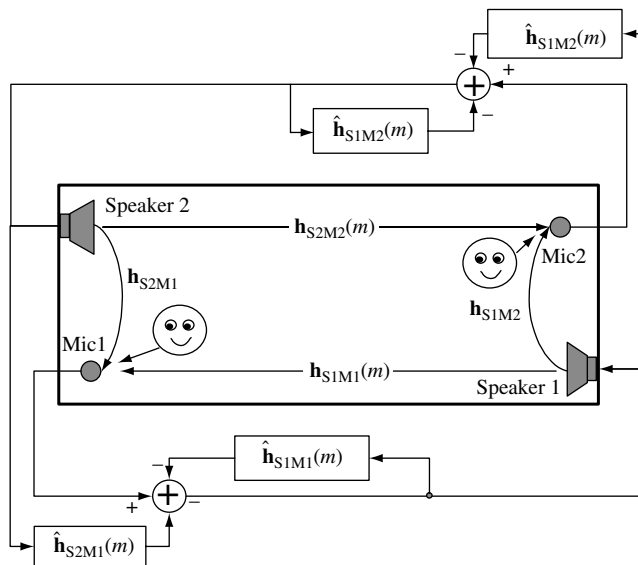


Figure 14.13 A block diagram illustration of a MIMO echo-cancellation in a car cabin communication system.

An interesting system solution to mitigate the effect of feedback in MIMO systems is to combine beam-forming microphone arrays with echo cancellation. Microphone arrays introduced in Chapters 1 and 15 form a beam where any signal arriving from directions outside the beam is attenuated. This would help reduce the feedback and can be used in combination with echo cancellers.

14.8 SUMMARY

Telephone line echo and acoustic feedback echo affect the functioning of telecommunications and teleconferencing systems. In general, line echo cancellation is a relatively less complex problem than acoustic echo cancellation because acoustic cancellers need to model the more complex environment of the space of a room.

We began this chapter with a study of the telephone line echoes arising from the mismatch at the two/four-wire hybrid bridge. In Section 14.2, line echo suppression and adaptive line echo cancellation were considered. For adaptation of an echo canceller, the LMS or the RLS adaptation methods can be used. The RLS methods provides a faster convergence rate and better overall performance at the cost of higher computational complexity.

In Section 14.3 we considered the acoustic coupling between a loudspeaker and a microphone system. Acoustic feedback echo can result in howling, and can disrupt the performance of teleconference, hands-free telephones and hearing aid systems. The main problems in implementation of acoustic echo cancellation systems are the requirement for a large filter to model the relatively long echo, and the adaptation problems associated with the eigenvalue spread of the signal. The sub-band echo canceller introduced in Section 14.4 alleviates these problems.

For stereophonic and MIMO systems the problem of acoustic echo cancellation remains an important and challenging research issue.

BIBLIOGRAPHY

- Allen J., Berkley D. and Blauret J. (1977) Multi-microphone signal processing technique to remove room reverberation from speech signals. *J. Acoust. Soc. Am.*, **62**: 4.
- Armbruster W. (1992) Wideband acoustic echo canceller with two filter structure. *Proc. Eusipco-92*, Vol. 3, pp. 1611–1617.
- Benesty J. Morgan D.R. Sondhi M.M. (1998) A better understanding of an improved solution to the problem of stereophonic acoustic echo cancellation, *IEEE Trans. Speech and Audio Processing*, Vol 6. No. 2 March, pp. 156–165.
- Carter G. (1987) Coherence and time delay estimation. *Proc. IEEE*, **75**(2): 236–255.
- Flanagan J.L., Johnston J.D., Zahn R. and Elko G. (1985) Computer-steered microphone arrays for sound transduction in large rooms. *J. Acoust. Soc. Ame.*, **78**: 1508–1518.
- Flanagan J.L., Berkley D., Elko G., West J. and Sondhi: M. (1991) Autodirective microphone systems. *Acoustica*, **73**: 58–71.
- Gao X.Y. and Snelgrove W.M. (1991) Adaptive linearisation of a loudspeaker. *IEEE Proceedings International Conference on Acoustics, Speech and Signal Processing, ICASSP-91*, Vol. 3, pp. 3589–3592.
- Gilloire A. and Vetterli M. (1994) Adaptive filtering in sub-bands with critical sampling: analysis, experiments and applications to acoustic echo cancellation. *IEEE. Trans. Signal Processing*, **40**: 320–328.
- Gritton C.W. and Lin D.W. (1984) Echo cancellation algorithms. *IEEE ASSP Mag.*, **1**(2): 30–37.
- Gustafsson S. and Martin R. (1997) Combined acoustic echo control and noise reduction for mobile communications. *Proc. EuroSpeech-97*, pp. 1403–1406.

- Hansler E. (1992) The hands-free telephone problem. An annotated bibliography. *Signal Process.*, **27**: 259–271.
- Hart J.E., Naylor P.A. and Tanrikulu O. (1993) Polyphase allpass IIR structures for subband acoustic echo cancellation. *EuroSpeech-93*, Vol. 3, pp. 1813–1816.
- Kellermann W. (1988) Analysis and design of multirate systems for cancellation of acoustical echoes. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-88*, pp. 2570–2573.
- Knapp M.E. (1992) Acoustic echo cancellation: performance and structures. M. Eng. Thesis, Carleton University, Ottawa.
- Martin R. and Altenhoner J. (1995) Coupled adaptive filters for acoustic echo control and noise reduction. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, vol. 5, pp. 3043–3046.
- McCaslin S. R., Hemkumar N. and Redheendran B. (1997) Double-talk detector for echo canceller. US patent no. 5631900, 20 May.
- Oslen H.F. (1964) *Acoustical Engineering*, T Van Nostrand, Toronto.
- Schroeder M.R. (1964) Improvement of acoustic-feedback stability by frequency shifting. *J. Acoust. Soc. Am.*, **36**: 1718–1724.
- Silverman H.F. and Kirtman E. (1992) A two-stage algorithm for determining talker location from linear microphone array data. *Comput. Speech Lang.*, **6**: 129–152.
- Sondhi M.M. (1967) An adaptive echo canceller. *Bell Syst. Tech. J.*, **46**: 497–511.
- Sondhi M.M. and Berkley D.A. (1980) Silencing echoes on the telephone network. *Proc. IEEE*, **68**: 948–963.
- Sondhi M.M. and Morgan D.R. (1991) Acoustic echo cancellation for stereophonic teleconferencing. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.
- Tanrikulu O., Baykal B., Constantinides A.G. and Chambes J.A. (1995) Finite-precision design and implementation of all-pass polyphase networks for echo cancellation in subbands. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing, ICASSP-95*, Vol. 5, pp. 3039–3042.
- Vaidyanathan P.P. (1993) *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ.
- Widrow B., Mccool J.M., Larimore M.G. and Johnson C.R. (1976) Stationary and nonstationary learning characteristics of the LMS adaptive filters. *Proc. IEEE*, **64**(8): 1151–1162.
- Ye H. and Wu B.-X. (1991) A new double-talk detection algorithm based on the orthogonality theorem. *IEEE Trans. Communi.*, **39**(11): 1542–1545.

15



Channel Equalisation and Blind Deconvolution

15.1 Introduction	372
15.2 Blind Equalisation Using the Channel Input Power Spectrum	379
15.3 Equalisation Based on Linear Prediction Models	382
15.4 Bayesian Blind Deconvolution and Equalisation	385
15.5 Blind Equalisation for Digital Communications Channels	393
15.6 Equalisation Based on Higher-order Statistics	398
15.7 Summary	406
Bibliography	406

Blind deconvolution is the process of unravelling two unknown signals that have been convolved. An important application of blind deconvolution is in blind equalisation for restoration of a signal distorted in transmission through a communication channel. Blind equalisation has a wide range of applications, for example in digital telecommunications for removal of inter-symbol interference, in speech recognition for removal of the effects of microphones and channels, in deblurring of distorted images, in dereverberation of acoustic recordings and in seismic data analysis.

In practice, blind equalisation is only feasible if some useful statistics of the channel input, and perhaps also of the channel itself, are available. The success of a blind equalisation method depends on how much is known about the statistics of the channel input, and how useful this knowledge is in the channel identification and equalisation process. This chapter begins with an introduction to the basic ideas of deconvolution and channel equalisation. We study blind equalisation based on the channel input power spectrum, equalisation through separation of the input signal and channel response models, Bayesian equalisation, nonlinear adaptive equalisation for digital communication channels and equalisation of maximum-phase channels using higher-order statistics.

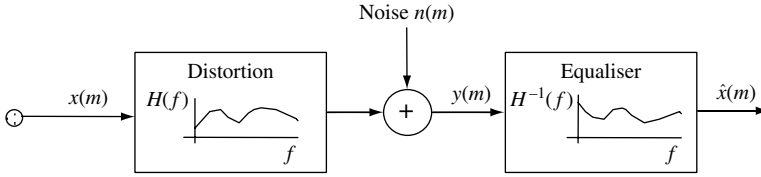


Figure 15.1 Illustration of a channel distortion model followed by an equaliser.

15.1 INTRODUCTION

In this chapter we consider the recovery of a signal distorted, in transmission through a channel, by a convolutional process and observed in additive noise. The process of recovery of a signal convolved with the impulse response of a communication channel, or a recording medium, is known as deconvolution or equalization. Figure 15.1 illustrates a typical model for a distorted and noisy signal, followed by an equaliser. Let $x(m)$, $n(m)$ and $y(m)$ denote the channel input, the channel noise and the observed channel output, respectively. The channel input/output relation can be expressed as

$$y(m) = h[x(m)] + n(m) \quad (15.1)$$

where the function $h[\cdot]$ is the channel distortion. In general, the channel response may be time-varying and nonlinear. In this chapter, it is assumed that the effects of a channel can be modelled using a stationary, or a slowly time-varying, linear transversal filter. For a linear transversal filter model of the channel, Equation (15.1) becomes

$$y(m) = \sum_{k=0}^{P-1} h_k(m)x(m-k) + n(m) \quad (15.2)$$

where $h_k(m)$ is the coefficient of a P th-order linear FIR filter model of the channel. For a time-invariant channel model, $h_k(m) = h_k$.

In the frequency domain, Equation (15.2) becomes

$$Y(f) = X(f)H(f) + N(f) \quad (15.3)$$

where $Y(f)$, $X(f)$, $H(f)$ and $N(f)$ are the frequency spectra of the channel output, the channel input, the channel response and the additive noise, respectively. Ignoring the noise term and taking the logarithm of Equation (15.3) yields

$$\ln |Y(f)| = \ln |X(f)| + \ln |H(f)| \quad (15.4)$$

From Equation (15.4), in the log-frequency domain the effect of channel distortion is the addition of a ‘tilt’ term, $\ln |H(f)|$, to the signal spectrum.

15.1.1 THE IDEAL INVERSE CHANNEL FILTER

The ideal inverse-channel filter, or the ideal equaliser, recovers the original input from the channel output signal. In the frequency domain, the ideal inverse channel filter can be expressed as

$$H(f)H^{\text{inv}}(f) = 1 \quad (15.5)$$

In Equation (15.5) $H^{\text{inv}}(f)$ is used to denote the inverse channel filter. For the ideal equaliser we have $H^{\text{inv}}(f) = H^{-1}(f)$, or, expressed in the log-frequency domain, $\ln H^{\text{inv}}(f) = -\ln H(f)$. The general form of Equation (15.5) is given by the z -transform relation

$$H(z)H^{\text{inv}}(z) = z^{-N} \quad (15.6)$$

for some value of the delay N that makes the channel inversion process causal. Taking the inverse Fourier transform of Equation (15.5), we have the following convolutional relation between the impulse responses of the channel $\{h_k\}$ and the ideal inverse channel response $\{h_k^{\text{inv}}\}$:

$$\sum_k h_k^{\text{inv}} h_{i-k} = \delta(i) \quad (15.7)$$

where $\delta(i)$ is the Kronecker delta function. Assuming the channel output is noise-free and the channel is invertible, the ideal inverse channel filter can be used to reproduce the channel input signal with zero error, as follows. The inverse filter output, $\hat{x}(m)$, with the distorted signal, $y(m)$, as the input, is given as

$$\begin{aligned} \hat{x}(m) &= \sum_k h_k^{\text{inv}} y(m-k) \\ &= \sum_k h_k^{\text{inv}} \sum_j h_j x(m-k-j) \\ &= \sum_i x(m-i) \sum_k h_k^{\text{inv}} h_{i-k} \end{aligned} \quad (15.8)$$

The last line of Equation (15.8) is derived by a change of variables $i = k + j$ in the second line and rearrangement of the terms. For the ideal inverse channel filter, substitution of Equation (15.7) in Equation (15.8) yields

$$\hat{x}(m) = \sum_i \delta(i)x(m-i) = x(m) \quad (15.9)$$

which is the desired result. In practice, it is not advisable to implement $H^{\text{inv}}(f)$ simply as $H^{-1}(f)$ because, in general, a channel response may be noninvertible. Even for invertible channels, a straightforward implementation of the inverse channel filter, $H^{-1}(f)$, can cause problems. For example, at frequencies where $H(f)$ is small, its inverse, $H^{-1}(f)$, is large, and this can lead to noise amplification if the signal-to-noise ratio is low.

15.1.2 EQUALISATION ERROR, CONVOLUTIONAL NOISE

The equalisation error signal, also called the convolutional noise, is defined as the difference between the channel equaliser output and the desired signal:

$$\begin{aligned} v(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \end{aligned} \quad (15.10)$$

where \hat{h}_k^{inv} is an estimate of the inverse channel filter. Assuming that there is an ideal equaliser, h_k^{inv} , that can recover the channel input signal $x(m)$ from the channel output $y(m)$, we have

$$x(m) = \sum_{k=0}^{P-1} h_k^{\text{inv}} y(m-k) \quad (15.11)$$

Substitution of Equation (15.11) in Equation (15.10) yields

$$\begin{aligned} v(m) &= \sum_{k=0}^{P-1} h_k^{\text{inv}} y(m-k) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \\ &= \sum_{k=0}^{P-1} \tilde{h}_k^{\text{inv}} y(m-k) \end{aligned} \quad (15.12)$$

where $\tilde{h}_k^{\text{inv}} = h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}$. The equalisation error signal, $v(m)$, may be viewed as the output of an error filter, \tilde{h}_k^{inv} , in response to the input $y(m-k)$, hence the name ‘convolutional noise’ for $v(m)$. When the equalisation process is proceeding well, such that $\hat{x}(m)$ is a good estimate of the channel input, $x(m)$, then the convolutional noise is relatively small and decorrelated and can be modelled as a zero mean Gaussian random process.

15.1.3 BLIND EQUALISATION

The equalisation problem is relatively simple when the channel response is known and invertible, and when the channel output is not noisy. However, in most practical cases, the channel response is unknown, time-varying, nonlinear, and may also be noninvertible. Furthermore, the channel output is often observed in additive noise.

Digital communication systems provide equaliser-training periods, during which a *training* pseudo-noise (PN) sequence, also available at the receiver, is transmitted. A synchronised version of the PN sequence is generated at the receiver, where the channel input and output signals are used for the identification of the channel equaliser, as illustrated in Figure 15.2(a). The obvious drawback of using training periods for channel equalization is that power, time and bandwidth are consumed for the equalisation process.

It is preferable to have a ‘blind’ equalisation scheme that can operate without access to the channel input, as illustrated in Figure 15.2(b). Furthermore, in some applications, such as the restoration of acoustic recordings, or blurred images, all that is available is the distorted signal and the only restoration method applicable is blind equalisation.

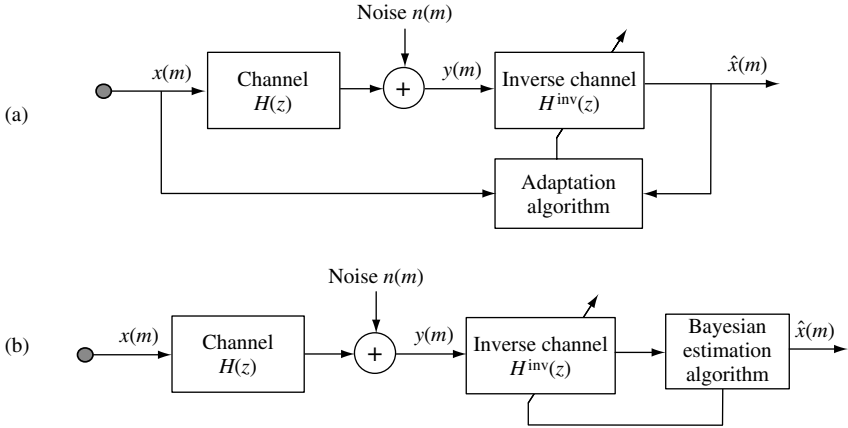


Figure 15.2 A comparative illustration of (a) a conventional equaliser with access to channel input and output, and (b) a blind equaliser.

Blind equalisation is feasible only if some statistical knowledge of the channel input, and perhaps that of the channel, is available. Blind equalisation involves two stages of channel identification, and deconvolution of the input signal and the channel response, as follows.

Channel identification – the general form of a channel estimator can be expressed as

$$\hat{\mathbf{h}} = \psi(\mathbf{y}, \mathcal{M}_x, \mathcal{M}_h) \tag{15.13}$$

where ψ is the channel estimator, the vector $\hat{\mathbf{h}}$ is an estimate of the channel response, \mathbf{y} is the channel output, and \mathcal{M}_x and \mathcal{M}_h are statistical models of the channel input and the channel response, respectively.

Channel identification methods rely on utilisation of a knowledge of the following characteristics of the input signal and the channel:

- (1) The distribution of the channel input signal – for example, in decision-directed channel equalisation, described in Section 15.5, the knowledge that the input is a binary signal is used in a binary decision device to estimate the channel input and to ‘direct’ the equalizer adaptation process.
- (2) The relative durations of the channel input and the channel impulse response – the duration of a channel impulse response is usually orders of magnitude smaller than that of the channel input. This observation is used in Section 15.3.1 to estimate a stationary channel from the long-time averages of the channel output.
- (3) The stationary, or time-varying characteristics of the input signal process and the channel – in Section 15.3.1, a method is described for the recovery of a nonstationary signal convolved with the impulse response of a stationary channel.

Channel equalisation – assuming that the channel is invertible, the channel input signal, $x(m)$, can be recovered using an inverse channel filter as

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{15.14}$$

In the frequency domain, Equation (15.14) becomes

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f)Y(f) \tag{15.15}$$

In practice, perfect recovery of the channel input may not be possible, either because the channel is noninvertible or because the output is observed in noise. A channel is noninvertible if:

- (1) The channel transfer function is maximum-phase – the transfer function of a maximum-phase channel has zeros outside the unit circle, and hence the inverse channel has unstable poles. Maximum-phase channels are considered in the next section.
- (2) The channel transfer function maps many inputs to the same output – in these situations, a stable closed-form equation for the inverse channel does not exist, and instead an iterative deconvolution method is used. Figure 15.3 illustrates the frequency response of a channel that has one invertible and two noninvertible regions. In the noninvertible regions, the signal frequencies are heavily attenuated and lost to channel noise. In the invertible region, the signal is distorted but recoverable. This example illustrates that the inverse filter must be implemented with care in order to avoid undesirable results, such as noise amplification at frequencies with low SNR.

15.1.4 MINIMUM- AND MAXIMUM-PHASE CHANNELS

For stability, all the poles of the transfer function of a channel must lie inside the unit circle. If all the zeros of the transfer function are also inside the unit circle then the channel is said to be a minimum-phase channel. If some of the zeros are outside the unit circle then the

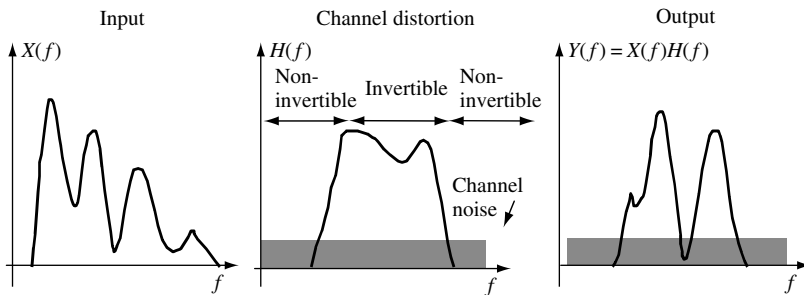


Figure 15.3 Illustration of the invertible and noninvertible regions of a channel.

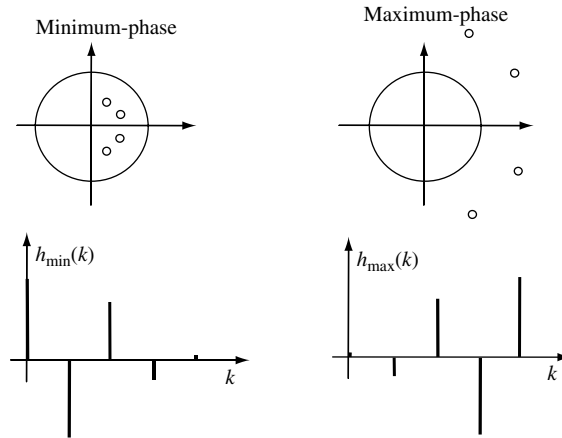


Figure 15.4 Illustration of the zero diagram and impulse response of fourth-order maximum-phase and minimum-phase FIR filters.

channel is said to be a maximum-phase channel. The inverse of a minimum-phase channel has all its poles inside the unit circle, and is therefore stable. The inverse of a maximum-phase channel has some of its poles outside the unit circle; therefore it has an exponentially growing impulse response and is unstable. However, a stable approximation of the inverse of a maximum-phase channel may be obtained by truncating the impulse response of the inverse filter. Figure 15.4 illustrates examples of maximum-phase and minimum-phase fourth-order FIR filters.

When both the channel input and output signals are available, in the correct synchrony, it is possible to estimate the channel magnitude and phase response using the conventional least square error criterion. In blind deconvolution, there is no access to the exact instantaneous value or the timing of the channel input signal. The only information available is the channel output and some statistics of the channel input. The second-order statistics of a signal (i.e. the correlation or the power spectrum) do not include the phase information; hence it is not possible to estimate the channel phase from the second-order statistics. Furthermore, the channel phase cannot be recovered if the input signal is Gaussian, because a Gaussian process of known mean is entirely specified by the autocovariance matrix, and autocovariance matrices do not include any phase information. For estimation of the phase of a channel, we can either use a nonlinear estimate of the desired signal to direct the adaptation of a channel equaliser as in Section 15.5, or we can use the higher-order statistics as in Section 15.6.

15.1.5 WIENER EQUALISER

In this section, we consider the least square error Wiener equalisation. Note that, in its conventional form, Wiener equalisation is not a form of blind equalisation, because the implementation of a Wiener equaliser requires the cross-correlation of the channel input and

output signals, which are not available in a blind equalisation application. The Wiener filter estimate of the channel input signal is given by

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \quad (15.16)$$

where \hat{h}_k^{inv} is an FIR Wiener filter estimate of the inverse channel impulse response. The equalisation error signal, $v(m)$, is defined as

$$v(m) = x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \quad (15.17)$$

The Wiener equaliser with input $y(m)$ and desired output $x(m)$ is obtained from Equation (6.10) as

$$\hat{\mathbf{h}}^{\text{inv}} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy} \quad (15.18)$$

where \mathbf{R}_{yy} is the $P \times P$ autocorrelation matrix of the channel output, and \mathbf{r}_{xy} is the P -dimensional cross-correlation vector of the channel input and output signals. A more expressive form of Equation (15.18) can be obtained by writing the noisy channel output signal in vector equation form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (15.19)$$

where \mathbf{y} is an N -sample channel output vector, \mathbf{x} is an $N + P$ -sample channel input vector including the P initial samples, \mathbf{H} is an $N \times (N + P)$ channel distortion matrix whose elements are composed of the coefficients of the channel filter, and \mathbf{n} is a noise vector. The autocorrelation matrix of the channel output can be obtained from Equation (15.19) as

$$\mathbf{R}_{yy} = \mathcal{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{H}\mathbf{R}_{xx}\mathbf{H}^T + \mathbf{R}_{nn} \quad (15.20)$$

where $\mathcal{E}[\cdot]$ is the expectation operator. The cross-correlation vector \mathbf{r}_{xy} of the channel input and output signals becomes

$$\mathbf{r}_{xy} = \mathcal{E}[\mathbf{x}\mathbf{y}] = \mathbf{H}\mathbf{r}_{xx} \quad (15.21)$$

Substitution of Equations (15.20) and (15.21) in Equation (15.18) yields the Wiener equaliser as

$$\hat{\mathbf{h}}^{\text{inv}} = (\mathbf{H}\mathbf{R}_{xx}\mathbf{H}^T + \mathbf{R}_{nn})^{-1} \mathbf{H}\mathbf{r}_{xx} \quad (15.22)$$

The derivation of the Wiener equaliser in the frequency domain is as follows. The Fourier transform of the equaliser output is given by

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f)Y(f) \quad (15.23)$$

where $Y(f)$ is the channel output and $\hat{H}^{\text{inv}}(f)$ is the frequency response of the Wiener equaliser. The error signal, $V(f)$, is defined as

$$\begin{aligned} V(f) &= X(f) - \hat{X}(f) \\ &= X(f) - \hat{H}^{\text{inv}}(f)Y(f) \end{aligned} \quad (15.24)$$

As in Section 6.5, minimisation of the expectation of the squared magnitude of $V(f)$ results in the frequency Wiener equaliser given by

$$\begin{aligned} \hat{H}^{\text{inv}}(f) &= \frac{P_{XY}(f)}{P_{YY}(f)} \\ &= \frac{P_{XX}(f)H^*(f)}{P_{XX}(f)|H(f)|^2 + P_{NN}(f)} \end{aligned} \quad (15.25)$$

where $P_{XX}(f)$ is the channel input power spectrum, $P_{NN}(f)$ is the noise power spectrum, $P_{XY}(f)$ is the cross-power spectrum of the channel input and output signals, and $H(f)$ is the frequency response of the channel. Note that, in the absence of noise, $P_{NN}(f) = 0$ and the Wiener inverse filter becomes $H^{\text{inv}}(f) = H^{-1}(f)$.

15.2 BLIND EQUALISATION USING THE CHANNEL INPUT POWER SPECTRUM

One of the early papers on blind deconvolution was by Stockham *et al.* (1975) on dereverberation of old acoustic recordings. Acoustic recorders, as illustrated in Figure 15.5, had a bandwidth of about 200 Hz to 4 kHz. However, the limited bandwidth, or even the additive noise or scratch noise pulses, are not considered the major causes of distortions of acoustic recordings. The main distortion on acoustic recordings is due to reverberations of the recording horn instrument. An acoustic recording can be modelled as the convolution



Figure 15.5 Illustration of the early acoustic recording process on a wax disc. Acoustic recordings were made by focusing the sound energy through a horn, via a sound box, diaphragm and stylus mechanism, onto a wax disc. The sound was distorted by reverberations of the horn.

of the input audio signal, $x(m)$, and the impulse response of a linear filter model of the recording instrument, $\{h_k\}$, as in Equation (15.2), reproduced here for convenience

$$y(m) = \sum_{k=0}^{P-1} h_k x(m-k) + n(m) \tag{15.26}$$

or in the frequency domain as

$$Y(f) = X(f)H(f) + N(f) \tag{15.27}$$

where $H(f)$ is the frequency response of a linear time-invariant model of the acoustic recording instrument, and $N(f)$ is an additive noise. Multiplying both sides of Equation (15.27) by their complex conjugates, and taking the expectation, we obtain

$$\mathcal{E}[Y(f)Y^*(f)] = \mathcal{E}\{[X(f)H(f) + N(f)][X(f)H(f) + N(f)]^*\} \tag{15.28}$$

Assuming the signal $X(f)$ and the noise $N(f)$ are uncorrelated, Equation (15.28) becomes

$$P_{YY}(f) = P_{XX}(f) |H(f)|^2 + P_{NN}(f) \tag{15.29}$$

where $P_{YY}(f)$, $P_{XX}(f)$ and $P_{NN}(f)$ are the power spectra of the distorted signal, the original signal and the noise, respectively. From Equation (15.29), an estimate of the spectrum of the channel response can be obtained as

$$|H(f)|^2 = \frac{P_{YY}(f) - P_{NN}(f)}{P_{XX}(f)} \tag{15.30}$$

In practice, Equation (15.30) is implemented using time-averaged estimates of the of the power spectra.

15.2.1 HOMOMORPHIC EQUALISATION

In homomorphic equalisation, the convolutional distortion is transformed, first into a multiplicative distortion through a Fourier transform of the distorted signal, and then into an additive distortion by taking the logarithm of the spectrum of the distorted signal. A further inverse Fourier transform operation converts the log-frequency variables into cepstral variables, as illustrated in Figure 15.6. Through homomorphic transformation convolution becomes addition, and equalisation becomes subtraction.

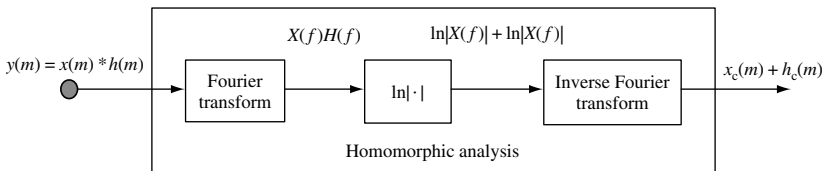


Figure 15.6 Illustration of homomorphic analysis in deconvolution.

Ignoring the additive noise term and transforming both sides of Equation (15.27) into log-spectral variables yields

$$\ln Y(f) = \ln X(f) + \ln H(f) \quad (15.31)$$

Note that in the log-frequency domain, the effect of channel distortion is the addition of a tilt to the spectrum of the channel input. Taking the expectation of Equation (15.31) yields

$$\mathcal{E}[\ln Y(f)] = \mathcal{E}[\ln X(f)] + \ln H(f) \quad (15.32)$$

In Equation (15.32), it is assumed that the channel is time-invariant; hence $\mathcal{E}[\ln H(f)] = \ln H(f)$. Using the relation $\ln z = \ln |z| + j\angle z$, the term $\mathcal{E}[\ln X(f)]$ can be expressed as

$$\mathcal{E}[\ln X(f)] = \mathcal{E}[\ln |X(f)|] + j\mathcal{E}[\angle X(f)] \quad (15.33)$$

The first term on the right-hand side of Equation (15.33), $\mathcal{E}[\ln |X(f)|]$, is nonzero, and represents the frequency distribution of the signal power in decibels, whereas the second term, $\mathcal{E}[\angle X(f)]$, is the expectation of the phase, and can be assumed to be zero. From Equation (15.32), the log-frequency spectrum of the channel can be estimated as

$$\ln H(f) = \mathcal{E}[\ln Y(f)] - \mathcal{E}[\ln X(f)] \quad (15.34)$$

In practice, when only a single record of a signal is available, the signal is divided into a number of segments, and the average signal spectrum is obtained over time across the segments. Assuming that the length of each segment is long compared with the duration of the channel impulse response, we can write an approximate convolutional relation for the i th signal segment as

$$y_i(m) \approx x_i(m) * h_i(m) \quad (15.35)$$

The segments are windowed, using a Hamming or a Hanning window, to reduce the spectral leakage due to end effects at the edges of the segment. Taking the complex logarithm of the Fourier transform of Equation (15.35) yields

$$\ln Y_i(f) = \ln X_i(f) + \ln H_i(f) \quad (15.36)$$

Taking the time averages over N segments of the distorted signal record yields

$$\frac{1}{N} \sum_{i=0}^{N-1} \ln Y_i(f) = \frac{1}{N} \sum_{i=0}^{N-1} \ln X_i(f) + \frac{1}{N} \sum_{i=0}^{N-1} \ln H_i(f) \quad (15.37)$$

Estimation of the channel response from Equation (15.37) requires the average log spectrum of the undistorted signal $X(f)$. In Stockham's method for restoration of acoustic records, the expectation of the signal spectrum is obtained from a modern recording of the same musical material as the acoustic recording. From Equation (15.37), the estimate of the logarithm of the channel is given by

$$\ln \hat{H}(f) = \frac{1}{N} \sum_{i=0}^{N-1} \ln Y_i(f) - \frac{1}{N} \sum_{i=0}^{N-1} \ln X_i^M(f) \quad (15.38)$$

where $X^M(f)$ is the spectrum of a modern recording. The equaliser can then be defined as

$$\ln H^{\text{inv}}(f) = \begin{cases} -\ln \hat{H}(f), & 200 \text{ Hz} \leq f \leq 4000 \text{ Hz} \\ -40 \text{ dB} & \text{otherwise} \end{cases} \quad (15.39)$$

In Equation (15.39), the inverse acoustic channel is implemented in the range 200–4000 Hz, where the channel is assumed to be invertible. Outside this range, the signal is dominated by noise, and the inverse filter is designed to attenuate the noisy signal.

15.2.2 HOMOMORPHIC EQUALISATION USING A BANK OF HIGH-PASS FILTERS

In the log-frequency domain, channel distortion may be eliminated using a bank of high-pass filters. Consider a time sequence of log-spectra of the output of a channel described as

$$\ln Y_t(f) = \ln X_t(f) + \ln H_t(f) \quad (15.40)$$

where $Y_t(f)$ and $X_t(f)$ are the channel input and output derived from a Fourier transform of the t th signal segment. From Equation (15.40), the effect of a time-invariant channel is to add a constant term, $\ln H(f)$, to each frequency component of the channel input $X_t(f)$, and the overall result is a time-invariant tilt of the log-frequency spectrum of the original signal. This observation suggests the use of a bank of narrow-band high-pass notch filters for the removal of the additive distortion term $\ln H(f)$. A simple first-order recursive digital filter with its notch at zero frequency is given by

$$\ln \hat{X}_t(f) = \alpha \ln \hat{X}_{t-1}(f) + \ln Y_t(f) - \ln Y_{t-1}(f) \quad (15.41)$$

where the parameter α controls the bandwidth of the notch at zero frequency. Note that the filter bank also removes any d.c. component of the signal $\ln X(f)$; for some applications, such as speech recognition, this is acceptable.

15.3 EQUALISATION BASED ON LINEAR PREDICTION MODELS

Linear prediction models, described in Chapter 8, are routinely used in applications such as seismic signal analysis and speech processing, for the modelling and identification of a minimum-phase channel. Linear prediction theory is based on two basic assumptions: that the channel is minimum-phase and that the channel input is a random signal. Standard linear prediction analysis can be viewed as a blind deconvolution method, because both the channel response and the channel input are unknown, and the only information is the channel output and the assumption that the channel input is random and hence has a flat power spectrum. In this section, we consider blind deconvolution using linear predictive models for the channel and its input. The channel input signal is modelled as

$$X(z) = E(z)A(z) \quad (15.42)$$

where $X(z)$ is the z -transform of the channel input signal, $A(z)$ is the z -transfer function of a linear predictive model of the channel input and $E(z)$ is the z -transform of a random excitation signal. Similarly, the channel output can be modelled by a linear predictive model $H(z)$ with input $X(z)$ and output $Y(z)$ as

$$Y(z) = X(z)H(z) \tag{15.43}$$

Figure 15.7 illustrates a cascade linear prediction model for a channel input process $X(z)$ and a channel response $H(z)$. The channel output can be expressed as

$$\begin{aligned} Y(z) &= E(z)A(z)H(z) \\ &= E(z)D(z) \end{aligned} \tag{15.44}$$

where

$$D(z) = A(z)H(z) \tag{15.45}$$

The z -transfer function of the linear prediction models of the channel input signal and the channel can be expanded as

$$A(z) = \frac{G_1}{1 - \sum_{k=1}^P a_k z^{-k}} = \frac{G_1}{\prod_{k=1}^P (1 - \alpha_k z^{-1})} \tag{15.46}$$

$$H(z) = \frac{G_2}{1 - \sum_{k=1}^Q b_k z^{-k}} = \frac{G_2}{\prod_{k=1}^Q (1 - \beta_k z^{-1})} \tag{15.47}$$

where $\{a_k, \alpha_k\}$ and $\{b_k, \beta_k\}$ are the coefficients and the poles of the linear prediction models for the channel, input signal and the channel, respectively. Substitution of Equations (15.46) and (15.47) in Equation (15.45) yields the combined input-channel model as

$$D(z) = \frac{G}{1 - \sum_{k=1}^{P+Q} d_k z^{-k}} = \frac{G}{\prod_{k=1}^{P+Q} (1 - \gamma_k z^{-1})} \tag{15.48}$$

The total number of poles of the combined model for the input signal and the channel is the sum of the poles of the input signal model and the channel model.

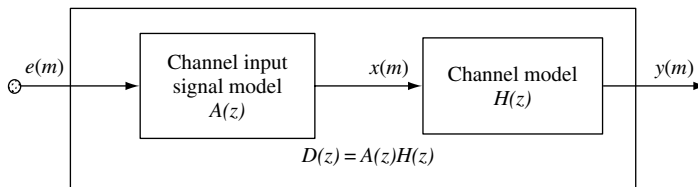


Figure 15.7 A distorted signal modelled as cascade of a signal model and a channel model.

15.3.1 BLIND EQUALISATION THROUGH MODEL FACTORISATION

A model-based approach to blind equalisation is to factorise the channel output model $D(z) = A(z)H(z)$ into a channel input signal model, $A(z)$, and a channel model, $H(z)$. If the channel input model, $A(z)$, and the channel model, $H(z)$, are nonfactorable then the only factors of $D(z)$ are $A(z)$ and $H(z)$. However, z -transfer functions are factorable into the roots, the so-called ‘poles and zeros’, of the models. One approach to model-based deconvolution is to factorise the model for the convolved signal into its poles and zeros, and classify the poles and zeros as either belonging to the signal or belonging to the channel.

Spencer and Rayner (1990) developed a method for blind deconvolution through factorisation of linear prediction models, based on the assumption that the channel is stationary with time-invariant poles whereas the input signal is nonstationary with time-varying poles. As an application, they considered the restoration of old acoustic recordings where a time-varying audio signal is distorted by the time-invariant frequency response of the recording equipment. For a simple example, consider the case when the signal and the channel are each modelled by a second-order linear predictive model. Let the time-varying second-order linear predictive model for the channel input signal $x(m)$ be

$$x(m) = a_1(m)x(m-1) + a_2(m)x(m-2) + G_1(m)e(m) \quad (15.49)$$

where $a_1(m)$ and $a_2(m)$ are the time-varying coefficients of the linear predictor model, $G_1(m)$ is the input gain factor and $e(m)$ is a zero-mean, unit variance, random signal. Now let $\alpha_1(m)$ and $\alpha_2(m)$ denote the time-varying poles of the predictor model of Equation (15.49); these poles are the roots of the polynomial

$$1 - a_1(m)z^{-1} - a_2(m)z^{-2} = [1 - z^{-1}\alpha_1(m)][1 - z^{-1}\alpha_2(m)] = 0 \quad (15.50)$$

Similarly, assume that the channel can be modelled by a second-order stationary linear predictive model as

$$y(m) = h_1y(m-1) + h_2y(m-2) + G_2x(m) \quad (15.51)$$

where h_1 and h_2 are the time-invariant predictor coefficients and G_2 is the channel gain. Let β_1 and β_2 denote the poles of the channel model; these are the roots of the polynomial

$$1 - h_1z^{-1} - h_2z^{-2} = (1 - z^{-1}\beta_1)(1 - z^{-1}\beta_2) = 0 \quad (15.52)$$

The combined cascade of the two second-order models of Equations (15.49) and (15.51) can be written as a fourth-order linear predictive model with input $e(m)$ and output $y(m)$:

$$y(m) = d_1(m)y(m-1) + d_2(m)y(m-2) + d_3(m)y(m-3) + d_4(m)y(m-4) + Ge(m) \quad (15.53)$$

where the combined gain $G = G_1G_2$. The poles of the fourth-order predictor model of Equation (15.53) are the roots of the following polynomial:

$$1 - d_1(m)z^{-1} - d_2(m)z^{-2} - d_3(m)z^{-3} - d_4(m)z^{-4} = [1 - z^{-1}\alpha_1(m)][1 - z^{-1}\alpha_2(m)][1 - z^{-1}\beta_1][1 - z^{-1}\beta_2] = 0 \quad (15.54)$$

In Equation (15.54) the poles of the fourth-order predictor are $\alpha_1(m)$, $\alpha_2(m)$, β_1 and β_2 . The above argument on factorisation of the poles of time-varying and stationary models can be generalised to a signal model of order P and a channel model of order Q .

In Spencer and Rayner, the separation of the stationary poles of the channel from the time-varying poles of the channel input is achieved through a clustering process. The signal record is divided into N segments and each segment is modelled by an all-pole model of order $P + Q$ where P and Q are the assumed to be model orders for the channel input and the channel, respectively. In all, there are $N(P + Q)$ values which are clustered to form $P + Q$ clusters. Even if both the signal and the channel were stationary, the poles extracted from different segments would have variations due to the random character of the signals from which the poles are extracted. Assuming that the variances of the estimates of the stationary poles are small compared with the variations of the time-varying poles, it is expected that, for each stationary pole of the channel, the N values extracted from N segments will form an N -point cluster of a relatively small variance. These clusters can be identified and the centre of each cluster taken as a pole of the channel model. This method assumes that the poles of the time-varying signal are well separated in space from the poles of the time-invariant signal.

15.4 BAYESIAN BLIND DECONVOLUTION AND EQUALISATION

The Bayesian inference method, described in Chapter 4, provides a general framework for inclusion of statistical models of the channel input and the channel response. In this section we consider the Bayesian equalisation method, and study the case where the channel input is modelled by a set of hidden Markov models. The Bayesian risk for a channel estimate $\hat{\mathbf{h}}$ is defined as

$$\begin{aligned} \mathcal{R}(\hat{\mathbf{h}}|\mathbf{y}) &= \int_{\mathbf{H}} \int_{\mathbf{X}} C(\hat{\mathbf{h}}, \mathbf{h}) f_{\mathbf{X}, \mathbf{H}|\mathbf{Y}}(\mathbf{x}, \mathbf{h}|\mathbf{y}) \, d\mathbf{x} \, d\mathbf{h} \\ &= \frac{1}{f_{\mathbf{Y}}(\mathbf{y})} \int_{\mathbf{H}} C(\hat{\mathbf{h}}, \mathbf{h}) f_{\mathbf{Y}|\mathbf{H}}(\mathbf{y}|\mathbf{h}) f_{\mathbf{H}}(\mathbf{h}) \, d\mathbf{h} \end{aligned} \tag{15.55}$$

where $C(\hat{\mathbf{h}}, \mathbf{h})$ is the cost of estimating the channel \mathbf{h} as $\hat{\mathbf{h}}$, $f_{\mathbf{X}, \mathbf{H}|\mathbf{Y}}(\mathbf{x}, \mathbf{h}|\mathbf{y})$ is the joint posterior density of the channel \mathbf{h} and the channel input \mathbf{x} , $f_{\mathbf{Y}|\mathbf{H}}(\mathbf{y}|\mathbf{h})$ is the observation likelihood and $f_{\mathbf{H}}(\mathbf{h})$ is the prior pdf of the channel. The Bayesian estimate is obtained by minimisation of the risk function, $\mathcal{R}(\hat{\mathbf{h}}|\mathbf{y})$. There are a variety of Bayesian-type solutions depending on the choice of the cost function and the prior knowledge, as described in Chapter 4.

In this section, it is assumed that the convolutional channel distortion is transformed into an additive distortion through transformation of the channel output into log-spectral or cepstral variables. Ignoring the channel noise, the relation between the cepstra of the channel input and output signals is given by

$$\mathbf{y}(m) = \mathbf{x}(m) + \mathbf{h} \tag{15.56}$$

where the cepstral vectors $\mathbf{x}(m)$, $\mathbf{y}(m)$ and \mathbf{h} are the channel input, the channel output and the channel, respectively.

15.4.1 CONDITIONAL MEAN CHANNEL ESTIMATION

A commonly used cost function in the Bayesian risk of Equation (15.55) is the mean square error, $C(\mathbf{h} - \hat{\mathbf{h}}) = \|\mathbf{h} - \hat{\mathbf{h}}\|^2$, which results in the conditional mean (CM) estimate defined as

$$\hat{\mathbf{h}}^{\text{CM}} = \int_{\mathbf{H}} \mathbf{h} f_{\mathbf{H}|\mathbf{Y}}(\mathbf{h}|\mathbf{y}) d\mathbf{h} \quad (15.57)$$

The posterior density of the channel input signal may be conditioned on an estimate of the channel vector, $\hat{\mathbf{h}}$, and expressed as $f_{X|Y,H}(\mathbf{x}|\mathbf{y}, \hat{\mathbf{h}})$. The conditional mean of the channel input signal given the channel output \mathbf{y} and an estimate of the channel $\hat{\mathbf{h}}$ is

$$\begin{aligned} \hat{\mathbf{x}}^{\text{CM}} &= \mathcal{E}[\mathbf{x}|\mathbf{y}, \hat{\mathbf{h}}] \\ &= \int_{\mathbf{X}} \mathbf{x} f_{X|Y,H}(\mathbf{x}|\mathbf{y}, \hat{\mathbf{h}}) d\mathbf{x} \end{aligned} \quad (15.58)$$

Equations (15.57) and (15.58) suggest a two-stage iterative method for channel estimation and the recovery of the channel input signal.

15.4.2 MAXIMUM-LIKELIHOOD CHANNEL ESTIMATION

The ML channel estimate is equivalent to the case when the Bayes cost function and the channel prior are uniform. Assuming that the channel input signal has a Gaussian distribution with mean vector μ_x and covariance matrix Σ_{xx} , the likelihood of a sequence of NP -dimensional channel output vectors $\{\mathbf{y}(m)\}$ given a channel input vector \mathbf{h} is

$$\begin{aligned} f_{Y|H}[\mathbf{y}(0), \dots, \mathbf{y}(N-1)|\mathbf{h}] &= \prod_{m=0}^{N-1} f_X[\mathbf{y}(m) - \mathbf{h}] \\ &= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2} |\Sigma_{xx}|^{1/2}} \\ &\quad \times \exp\{[\mathbf{y}(m) - \mathbf{h} - \mu_x]^T \Sigma_{xx}^{-1} [\mathbf{y}(m) - \mathbf{h} - \mu_x]\} \end{aligned} \quad (15.59)$$

To obtain the ML estimate of the channel \mathbf{h} , the derivative of the log likelihood function $\ln f_Y(\mathbf{y}|\mathbf{h})$ with respect to \mathbf{h} is set to zero to yield

$$\hat{\mathbf{h}}^{\text{ML}} = \frac{1}{N} \sum_{m=0}^{N-1} [\mathbf{y}(m) - \mu_x] \quad (15.60)$$

15.4.3 MAXIMUM A POSTERIORI CHANNEL ESTIMATION

The MAP estimate, like the ML estimate, is equivalent to a Bayesian estimator with a uniform cost function. However, the MAP estimate includes the prior pdf of the channel. The prior pdf can be used to confine the channel estimate within a desired subspace of the

parameter space. Assuming that the channel input vectors are statistically independent, the posterior pdf of the channel given the observation sequence $\mathbf{Y} = \{\mathbf{y}(0), \dots, \mathbf{y}(N-1)\}$ is

$$\begin{aligned} f_{H|Y}[\mathbf{h}|\mathbf{y}(0), \dots, \mathbf{y}(N-1)] &= \prod_{m=0}^{N-1} \frac{1}{f_Y[\mathbf{y}(m)]} f_{Y|H}[\mathbf{y}(m)|\mathbf{h}] f_H(\mathbf{h}) \\ &= \prod_{m=0}^{N-1} \frac{1}{f_Y[\mathbf{y}(m)]} f_X[\mathbf{y}(m) - \mathbf{h}] f_H(\mathbf{h}) \end{aligned} \quad (15.61)$$

Assuming that the channel input, $\mathbf{x}(m)$, is Gaussian, $f_X[\mathbf{x}(m)] = \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$, with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the channel \mathbf{h} is also Gaussian, $f_H(\mathbf{h}) = \mathcal{N}(\mathbf{h}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_{hh})$, with mean vector $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Sigma}_{hh}$, the logarithm of the posterior pdf is

$$\begin{aligned} \ln f_{H|Y}[\mathbf{h}|\mathbf{y}(0), \dots, \mathbf{y}(N-1)] &= - \sum_{m=0}^{N-1} \ln f[\mathbf{y}(m)] - NP \ln(2\pi) - \frac{1}{2} \ln(|\boldsymbol{\Sigma}_{xx}| |\boldsymbol{\Sigma}_{hh}|) \\ &\quad - \sum_{m=0}^{N-1} \frac{1}{2} \{ [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_x]^T \boldsymbol{\Sigma}_{xx}^{-1} [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_x] + (\mathbf{h} - \boldsymbol{\mu}_h)^T \boldsymbol{\Sigma}_{hh}^{-1} (\mathbf{h} - \boldsymbol{\mu}_h) \} \end{aligned} \quad (15.62)$$

The MAP channel estimate, obtained by setting the derivative of the log posterior function $\ln f_{H|Y}(\mathbf{h}|\mathbf{y})$ to zero, is

$$\hat{\mathbf{h}}^{\text{MAP}} = (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{hh} (\bar{\mathbf{y}} - \boldsymbol{\mu}_x) + (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{xx} \boldsymbol{\mu}_h \quad (15.63)$$

where

$$\bar{\mathbf{y}} = \frac{1}{N} \sum_{m=0}^{N-1} \mathbf{y}(m) \quad (15.64)$$

is the time-averaged estimate of the mean of the observation vector. Note that, for a Gaussian process, the MAP and conditional mean estimates are identical.

15.4.4 CHANNEL EQUALISATION BASED ON HIDDEN MARKOV MODELS

This section considers blind deconvolution in applications where the statistics of the channel input are modelled by a set of hidden Markov models. An application of this method, illustrated in Figure 15.8, is in recognition of speech distorted by a communication channel or a microphone. A hidden Markov model is a finite-state Bayesian model, with a Markovian state prior and a Gaussian observation likelihood (see Chapter 5). An N -state HMM can be used to model a nonstationary process, such as speech, as a chain of N stationary states connected by a set of Markovian state transitions. The likelihood of an HMM \mathcal{M}_i and a sequence of NP -dimensional channel input vectors, $\mathbf{X} = [\mathbf{x}(0), \dots, \mathbf{x}(N-1)]$, can be expressed in terms of the state transition and the observation pdfs of \mathcal{M}_i as

$$f_{X|\mathcal{M}}(\mathbf{X}|\mathcal{M}_i) = \sum_{\mathbf{s}} f_{X|\mathcal{M},\mathbf{s}}(\mathbf{X}|\mathcal{M}_i, \mathbf{s}) P_{S|\mathcal{M}}(\mathbf{s}|\mathcal{M}_i) \quad (15.65)$$

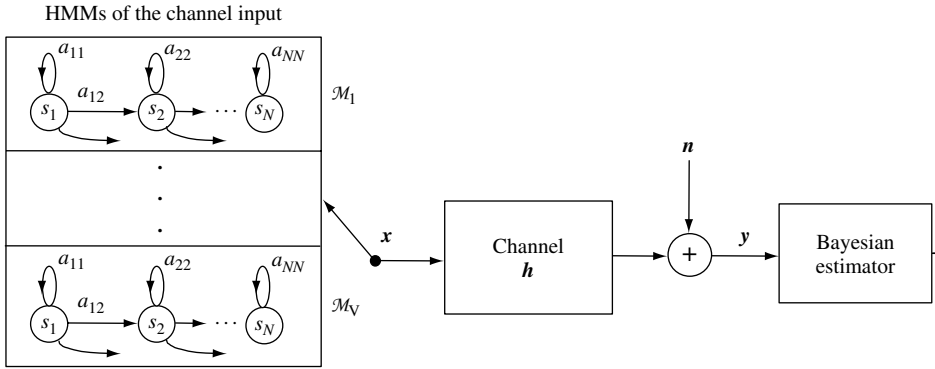


Figure 15.8 Illustration of a channel with the input modelled by a set of HMMs.

where $f_{X|\mathcal{M}_i, s}(X|\mathcal{M}_i, s)$ is the likelihood that the sequence $X = [x(0), \dots, x(N - 1)]$ was generated by the state sequence $s = [s(0), \dots, s(N - 1)]$ of the model \mathcal{M}_i , and $P_{s|\mathcal{M}_i}(s|\mathcal{M}_i)$ is the Markovian prior pmf of the state sequence s . The Markovian prior entails that the probability of a transition to the state i at time m depends only on the state at time $m - 1$ and is independent of the previous states. The transition probability of a Markov process is defined as

$$a_{ij} = P[s(m) = j | s(m - 1) = i] \tag{15.66}$$

where a_{ij} is the probability of making a transition from state i to state j . The HMM state observation probability is often modelled by a multivariate Gaussian pdf as

$$f_{X|\mathcal{M}_i, s}(x|\mathcal{M}_i, s) = \frac{1}{(2\pi)^{P/2} |\Sigma_{xx, s}|^{1/2}} \exp \left\{ -\frac{1}{2} [x - \mu_{x, s}]^T \Sigma_{xx, s}^{-1} [x - \mu_{x, s}] \right\} \tag{15.67}$$

where $\mu_{x, s}$ and $\Sigma_{xx, s}$ are the mean vector and the covariance matrix of the Gaussian observation pdf of the HMM state s of the model \mathcal{M}_i .

The HMM-based channel equalization problem can be stated as follows: given a sequence of NP -dimensional channel output vectors $Y = [y(0), \dots, y(N - 1)]$, and the prior knowledge that the channel input sequence is drawn from a set of V HMMs $\mathcal{M} = \{\mathcal{M}_i | i = 1, \dots, V\}$, estimate the channel response and the channel input.

The joint posterior pdf of an input word \mathcal{M}_i and the channel vector h can be expressed as

$$f_{\mathcal{M}, H|Y}(\mathcal{M}_i, h|Y) = P_{\mathcal{M}|H, Y}(\mathcal{M}_i|h, Y) f_{H|Y}(h|Y) \tag{15.68}$$

Simultaneous joint estimation of the channel vector h and classification of the unknown input word \mathcal{M}_i is a nontrivial exercise. The problem is usually approached iteratively by making an estimate of the channel response, and then using this estimate to obtain the channel input as

follows. From Bayes's rule, the posterior pdf of the channel \mathbf{h} conditioned on the assumption that the input model is \mathcal{M}_i and given the observation sequence \mathbf{Y} can be expressed as

$$f_{\mathbf{H}|\mathcal{M},\mathbf{Y}}(\mathbf{h}|\mathcal{M}_i, \mathbf{Y}) = \frac{1}{f_{\mathbf{Y}|\mathcal{M}}(\mathbf{Y}|\mathcal{M}_i)} f_{\mathbf{Y}|\mathcal{M},\mathbf{H}}(\mathbf{Y}|\mathcal{M}_i, \mathbf{h}) f_{\mathbf{H}|\mathcal{M}}(\mathbf{h}|\mathcal{M}_i) \quad (15.69)$$

The likelihood of the observation sequence, given the channel and the input word model, can be expressed as

$$f_{\mathbf{Y}|\mathcal{M},\mathbf{H}}(\mathbf{Y}|\mathcal{M}_i, \mathbf{h}) = f_{\mathbf{X}|\mathcal{M}}(\mathbf{Y} - \mathbf{h}|\mathcal{M}_i) \quad (15.70)$$

where it is assumed that the channel output is transformed into cepstral variables so that the channel distortion is additive. For a given input model \mathcal{M}_i , and state sequence $\mathbf{s} = [s(0), s(1), \dots, s(N-1)]$, the pdf of a sequence of N independent observation vectors $\mathbf{Y} = [\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(N-1)]$ is

$$\begin{aligned} f_{\mathbf{Y}|\mathbf{H},\mathbf{S},\mathcal{M}}(\mathbf{Y}|\mathbf{h}, \mathbf{s}, \mathcal{M}_i) &= \prod_{m=0}^{N-1} f_{\mathbf{X}|\mathbf{S},\mathcal{M}}[\mathbf{y}(m) - \mathbf{h}|s(m), \mathcal{M}_i] \\ &= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2}} |\boldsymbol{\Sigma}_{\mathbf{xx},s(m)}|^{1/2} \exp \left\{ -\frac{1}{2} [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_{\mathbf{x},s(m)}]^T \boldsymbol{\Sigma}_{\mathbf{xx},s(m)}^{-1} \right. \\ &\quad \left. [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_{\mathbf{x},s(m)}] \right\} \end{aligned} \quad (15.71)$$

Taking the derivative of the log-likelihood of Equation (15.71) with respect to the channel vector \mathbf{h} yields a maximum likelihood channel estimate as

$$\hat{\mathbf{h}}^{\text{ML}}(\mathbf{Y}, \mathbf{s}) = \sum_{m=0}^{N-1} \left(\sum_{k=0}^{N-1} \boldsymbol{\Sigma}_{\mathbf{xx},s(k)}^{-1} \right)^{-1} \boldsymbol{\Sigma}_{\mathbf{xx},s(m)}^{-1} [\mathbf{y}(m) - \boldsymbol{\mu}_{\mathbf{x},s(m)}] \quad (15.72)$$

Note that, when all the state observation covariance matrices are identical, the channel estimate becomes

$$\hat{\mathbf{h}}^{\text{ML}}(\mathbf{Y}, \mathbf{s}) = \frac{1}{N} \sum_{m=0}^{N-1} [\mathbf{y}(m) - \boldsymbol{\mu}_{\mathbf{x},s(m)}] \quad (15.73)$$

The ML estimate of Equation (15.73) is based on the ML state sequence, \mathbf{s} , of \mathcal{M}_i . In the following section we consider the conditional mean estimate over all state sequences of a model.

15.4.5 MAP CHANNEL ESTIMATE BASED ON HMMS

The conditional pdf of a channel \mathbf{h} averaged over all HMMs can be expressed as

$$f_{\mathbf{H}|\mathbf{Y}}(\mathbf{h}|\mathbf{Y}) = \sum_{i=1}^V \sum_{\mathbf{s}} f_{\mathbf{H}|\mathbf{Y},\mathbf{S},\mathcal{M}}(\mathbf{h}|\mathbf{Y}, \mathbf{s}, \mathcal{M}_i) P_{\mathbf{S}|\mathcal{M}_i}(s|\mathcal{M}_i) P_{\mathcal{M}}(\mathcal{M}_i) \quad (15.74)$$

where $P_{\mathcal{M}}(\mathcal{M}_i)$ is the prior pmf of the input words. Given a sequence of NP -dimensional observation vectors, $\mathbf{Y} = [\mathbf{y}(0), \dots, \mathbf{y}(N-1)]$, the posterior pdf of the channel \mathbf{h} along a state sequence \mathbf{s} of an HMM \mathcal{M}_i is defined as

$$\begin{aligned} f_{\mathbf{Y}|\mathbf{H},\mathbf{S},\mathcal{M}}(\mathbf{h}|\mathbf{Y}, \mathbf{s}, \mathcal{M}_i) &= \frac{1}{f_{\mathbf{Y}}(\mathbf{Y})} f_{\mathbf{Y}|\mathbf{H},\mathbf{S},\mathcal{M}}(\mathbf{Y}|\mathbf{h}, \mathbf{s}, \mathcal{M}_i) f_{\mathbf{H}}(\mathbf{h}) \\ &= \frac{1}{f_{\mathbf{Y}}(\mathbf{Y})} \prod_{m=0}^{N-1} \frac{1}{(2\pi)^P |\boldsymbol{\Sigma}_{\mathbf{xx},s(m)}|^{1/2} |\boldsymbol{\Sigma}_{\mathbf{hh}}|^{1/2}} \\ &\quad \times \exp \left\{ -\frac{1}{2} [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_{\mathbf{x},s(m)}]^T \boldsymbol{\Sigma}_{\mathbf{xx},s(m)}^{-1} [\mathbf{y}(m) - \mathbf{h} - \boldsymbol{\mu}_{\mathbf{x},s(m)}] \right\} \\ &\quad \times \exp \left[-\frac{1}{2} (\mathbf{h} - \boldsymbol{\mu}_{\mathbf{h}})^T \boldsymbol{\Sigma}_{\mathbf{hh}}^{-1} (\mathbf{h} - \boldsymbol{\mu}_{\mathbf{h}}) \right] \end{aligned} \quad (15.75)$$

where it is assumed that each state of the HMM has a Gaussian distribution with mean vector $\boldsymbol{\mu}_{\mathbf{x},s(m)}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{xx},s(m)}$, and that the channel \mathbf{h} is also Gaussian-distributed, with mean vector $\boldsymbol{\mu}_{\mathbf{h}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{hh}}$. The MAP estimate along state \mathbf{s} , on the left-hand side of Equation (15.75), can be obtained as

$$\begin{aligned} \hat{\mathbf{h}}^{\text{MAP}}(\mathbf{Y}, \mathbf{s}, \mathcal{M}_i) &= \sum_{m=0}^{N-1} \left[\sum_{k=0}^{N-1} \left(\boldsymbol{\Sigma}_{\mathbf{xx},s(k)}^{-1} + \boldsymbol{\Sigma}_{\mathbf{hh}}^{-1} \right) \right]^{-1} \boldsymbol{\Sigma}_{\mathbf{xx},s(m)}^{-1} [\mathbf{y}(m) - \boldsymbol{\mu}_{\mathbf{x},s(m)}] \\ &\quad + \left[\sum_{k=0}^{N-1} \left(\boldsymbol{\Sigma}_{\mathbf{xx},s(k)}^{-1} + \boldsymbol{\Sigma}_{\mathbf{hh}}^{-1} \right) \right]^{-1} \boldsymbol{\Sigma}_{\mathbf{hh}}^{-1} \boldsymbol{\mu}_{\mathbf{h}} \end{aligned} \quad (15.76)$$

The MAP estimate of the channel over all state sequences of all HMMs can be obtained as

$$\hat{\mathbf{h}}(\mathbf{Y}) = \sum_{i=1}^V \sum_{\mathbf{s}} \hat{\mathbf{h}}^{\text{MAP}}(\mathbf{Y}, \mathbf{s}, \mathcal{M}_i) P_{\mathbf{S}|\mathcal{M}}(\mathbf{s}|\mathcal{M}_i) P_{\mathcal{M}}(\mathcal{M}_i) \quad (15.77)$$

15.4.6 IMPLEMENTATIONS OF HMM-BASED DECONVOLUTION

In this section, we consider three implementation methods for HMM-based channel equalization.

Method I: Use of the Statistical Averages Taken Over All HMMs

A simple approach to blind equalisation, similar to that proposed by Stockham, is to use as the channel input statistics the average of the mean vectors and the covariance matrices, taken over all the states of all the HMMs, as

$$\boldsymbol{\mu}_{\mathbf{x}} = \frac{1}{VN_s} \sum_{i=1}^V \sum_{j=1}^{N_s} \boldsymbol{\mu}_{\mathcal{M}_i,j}, \quad \boldsymbol{\Sigma}_{\mathbf{xx}} = \frac{1}{VN_s} \sum_{i=1}^V \sum_{j=1}^{N_s} \boldsymbol{\Sigma}_{\mathcal{M}_i,j} \quad (15.78)$$

where $\boldsymbol{\mu}_{\mathcal{M}_i,j}$ and $\boldsymbol{\Sigma}_{\mathcal{M}_i,j}$ are the mean and the covariance of the j th state of the i th HMM, and V and N_s denote the number of models and number of states per model, respectively. The maximum likelihood estimate of the channel, $\hat{\mathbf{h}}^{\text{ML}}$, is defined as

$$\hat{\mathbf{h}}^{\text{ML}} = (\bar{\mathbf{y}} - \boldsymbol{\mu}_x) \tag{15.79}$$

where $\bar{\mathbf{y}}$ is the time-averaged channel output. The estimate of the channel input is

$$\hat{\mathbf{x}}(m) = \mathbf{y}(m) - \hat{\mathbf{h}}^{\text{ML}} \tag{15.80}$$

Using the averages over all states and models, the MAP channel estimate becomes

$$\hat{\mathbf{h}}^{\text{MAP}}(\mathbf{Y}) = \sum_{m=0}^{N-1} (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{hh} [\mathbf{y}(m) - \boldsymbol{\mu}_x] + (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{xx} \boldsymbol{\mu}_h \tag{15.81}$$

Method II: Hypothesised-input HMM Equalisation

In this method, for each candidate HMM in the input vocabulary, a channel estimate is obtained and then used to equalise the channel output, prior to the computation of a likelihood score for the HMM. Thus a channel estimate $\hat{\mathbf{h}}_w$ is based on the hypothesis that the input word is w . It is expected that a better channel estimate is obtained from the correctly hypothesised HMM, and a poorer estimate from an incorrectly hypothesised HMM. The hypothesised-input HMM algorithm is as follows (Figure 15.9):

for $i = 1$ to number of words V

- Step 1: using each HMM, \mathcal{M}_i , make an estimate of the channel, $\hat{\mathbf{h}}_i$;
- Step 2: using the channel estimate, $\hat{\mathbf{h}}_i$, estimate the channel input $\hat{\mathbf{x}}(m) = \mathbf{y}(m) - \hat{\mathbf{h}}_i$;
- Step 3: compute a probability score for model \mathcal{M}_i , given the estimate $[\hat{\mathbf{x}}(m)]$.

Select the channel estimate associated with the most probable word.

Figure 15.10 shows the ML channel estimates of two channels using unweighted average and hypothesised-input methods.

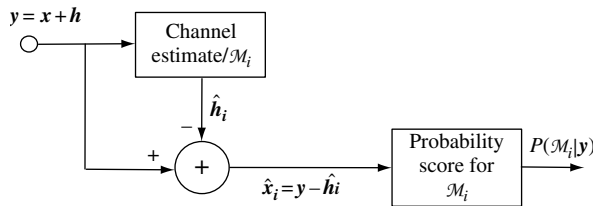


Figure 15.9 Hypothesised channel estimation procedure.

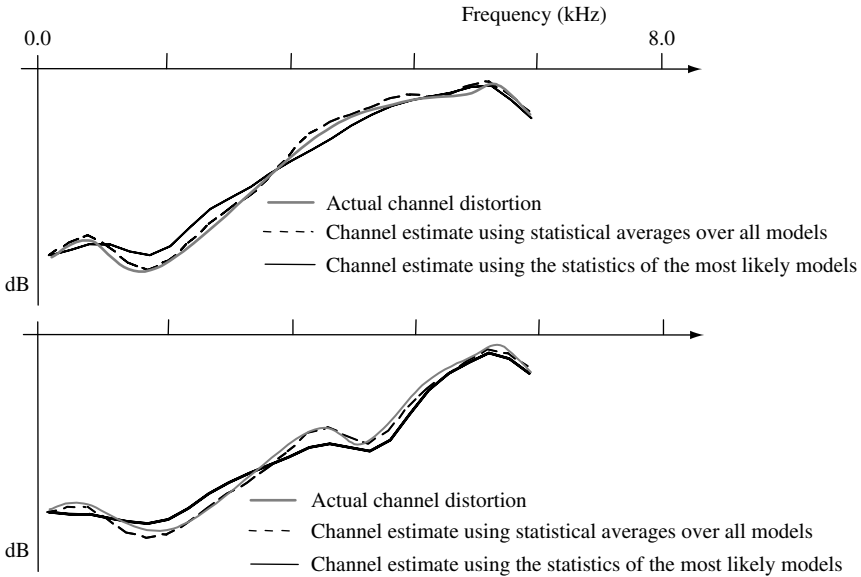


Figure 15.10 Illustration of actual and estimated channel response for two channels.

Method III: Decision-directed Equalisation

Blind adaptive equalisers are often composed of two distinct sections: an adaptive linear equaliser followed by a nonlinear estimator to improve the equaliser output. The output of the nonlinear estimator is the final estimate of the channel input, and is used as the desired signal to direct the equaliser adaptation. The use of the output of the nonlinear estimator as the desired signal assumes that the linear equalisation filter removes a large part of the channel distortion, thereby enabling the nonlinear estimator to produce an accurate estimate of the channel input. A method of ensuring that the equaliser locks into, and cancels, a large part of the channel distortion is to use a startup, equaliser training period during which a known signal is transmitted.

Figure 15.11 illustrates a blind equaliser incorporating an adaptive linear filter followed by an HMM classifier/estimator. The HMM classifies the output of the filter as one of a

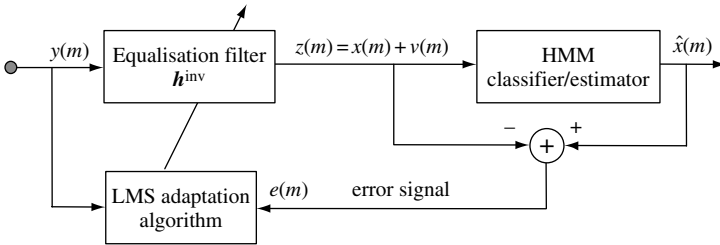


Figure 15.11 A decision-directed equaliser.

number of likely signals and provides an enhanced output, which is also used for adaptation of the linear filter. The output of the equaliser, $z(m)$, is expressed as the sum of the input to the channel, $x(m)$, and a so-called 'convolutional noise term', $v(m)$ as

$$z(m) = x(m) + v(m) \quad (15.82)$$

The HMM may incorporate state-based Wiener filters for suppression of the convolutional noise, $v(m)$, as described in Section 5.5. Assuming that the LMS adaptation method is employed, the adaptation of the equaliser coefficient vector is governed by the following recursive equation:

$$\hat{h}^{\text{inv}}(m) = \hat{h}^{\text{inv}}(m-1) + \mu e(m)y(m) \quad (15.83)$$

where $\hat{h}^{\text{inv}}(m)$ is an estimate of the optimal inverse channel filter, μ is an adaptation step size and the error signal $e(m)$ is defined as

$$e(m) = \hat{x}^{\text{HMM}}(m) - z(m) \quad (15.84)$$

where $\hat{x}^{\text{HMM}}(m)$ is the output of the HMM-based estimator and is used as the correct estimate of the desired signal to direct the adaptation process.

15.5 BLIND EQUALISATION FOR DIGITAL COMMUNICATIONS CHANNELS

High-speed transmission of digital data over analogue channels, such as telephone lines or a radio channel, requires adaptive equalisation to reduce decoding errors caused by channel distortions. In telephone lines, the channel distortions are due to the nonideal magnitude response and the nonlinear phase response of the lines. In radio channel environments, the distortions are due to nonideal channel response as well as the effects of multipath propagation of the radio waves via a multitude of different routes with different attenuations and delays. In general, the main types of distortions suffered by transmitted symbols are amplitude distortion, time dispersion and fading. Of these, time dispersion is perhaps the most important, and has received a great deal of attention. Time dispersion has the effect of smearing and elongating the duration of each symbol. In high-speed communication systems, where the data symbols closely follow each other, time dispersion results in an overlap of successive symbols, an effect known as inter-symbol interference (ISI), illustrated in Figure 15.12.

In a digital communications system, the transmitter modem takes N bits of binary data at a time, and encodes them into one of 2^N analogue symbols for transmission, at the signalling rate, over an analogue channel. At the receiver the analogue signal is sampled and decoded into the required digital format. Most digital modems are based on multilevel phase-shift keying, or combined amplitude and phase-shift keying schemes. In this section we consider multilevel pulse amplitude modulation (M -ary PAM) as a convenient scheme for the study of adaptive channel equalisation.

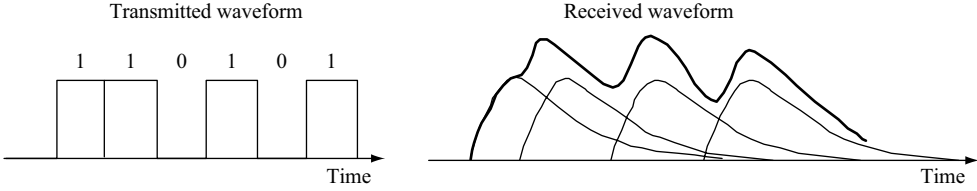


Figure 15.12 Illustration of intersymbol interference in a binary pulse amplitude modulation system.

Assume that at the transmitter modem, the k th set of N binary digits, is mapped into a pulse of duration T_s seconds and an amplitude $a(k)$. Thus the modulator output signal, which is the input to the communication channel, is given as

$$x(t) = \sum_k a(k)r(t - kT_s) \quad (15.85)$$

where $r(t)$ is a pulse of duration T_s and with an amplitude $a(k)$ that can assume one of $M = 2^N$ distinct levels. Assuming that the channel is linear, the channel output can be modelled as the convolution of the input signal and channel response:

$$y(t) = \int_{-\infty}^{\infty} h(\tau)x(t - \tau) d\tau \quad (15.86)$$

where $h(t)$ is the channel impulse response. The sampled version of the channel output is given by the following discrete-time equation:

$$y(m) = \sum_k h_k x(m - k) \quad (15.87)$$

To remove the channel distortion, the sampled channel output $y(m)$ is passed to an equaliser with an impulse response \hat{h}_k^{inv} . The equaliser output $z(m)$ is given as

$$\begin{aligned} z(m) &= \sum_k \hat{h}_k^{\text{inv}} y(m - k) \\ &= \sum_j x(m - j) \sum_k \hat{h}_k^{\text{inv}} h_{j-k} \end{aligned} \quad (15.88)$$

where Equation (15.87) is used to obtain the second line of Equation (15.88). The ideal equaliser output is $z(m) = x(m - D) = a(m - D)$ for some delay D that depends on the channel response and the length of the equaliser. From Equation (15.88), the channel distortion would be cancelled if

$$h_m^c = h_m * \hat{h}_m^{\text{inv}} = \delta(m - D) \quad (15.89)$$

where h_m^c is the combined impulse response of the cascade of the channel and the equaliser. A particular form of channel equaliser, for the elimination of ISI, is the Nyquist *zero-forcing* filter, where the impulse response of the combined channel and equaliser is defined as

$$h^c(kT_s + D) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (15.90)$$

Note that, in Equation (15.90), at the sampling instants the channel distortion is cancelled, and hence there is no ISI at the sampling instants. A function that satisfies Equation (15.90) is the sinc function, $h^c(t) = \sin(\pi f_s t) / \pi f_s t$, where $f_s = 1/T_s$. Zero-forcing methods are sensitive to deviations of $h^c(t)$ from the requirement of Equation (15.90), and also to jitters in the synchronisation and the sampling process.

15.5.1 LMS BLIND EQUALISATION

In this section, we consider the more general form of the LMS-based adaptive equaliser followed by a nonlinear estimator. In a conventional sample-adaptive filter, the filter coefficients are adjusted to minimise the mean squared distance between the filter output and the desired signal. In blind equalisation, the desired signal (which is the channel input) is not available. The use of an adaptive filter for blind equalisation requires an internally generated desired signal, as illustrated in Figure 15.13. Digital blind equalisers are composed of two distinct sections: an adaptive equaliser that removes a large part of the channel distortion, followed by a nonlinear estimator for an improved estimate of the channel input. The output of the nonlinear estimator is the final estimate of the channel input, and is used as the desired signal to direct the equalizer adaptation. A method of ensuring that the equalizer removes a large part of the channel distortion is to use a start-up, equaliser-training period during which a known signal is transmitted.

Assuming that the LMS adaptation method is employed, the adaptation of the equaliser coefficient vector is governed by the following recursive equation:

$$\hat{\mathbf{h}}^{inv}(m) = \hat{\mathbf{h}}^{inv}(m - 1) + \mu e(m)\mathbf{y}(m) \tag{15.91}$$

where $\hat{\mathbf{h}}^{inv}(m)$ is an estimate of the optimal inverse channel filter \mathbf{h}^{inv} , the scalar μ is the adaptation step size, and the error signal $e(m)$ is defined as

$$\begin{aligned} e(m) &= \psi[z(m)] - z(m) \\ &= \hat{x}(m) - z(m) \end{aligned} \tag{15.92}$$

where $\hat{x}(m) = \psi[z(m)]$ is a nonlinear estimate of the channel input. For example, in a binary communication system with an input alphabet $\{\pm a\}$, we can use a signum nonlinearity such

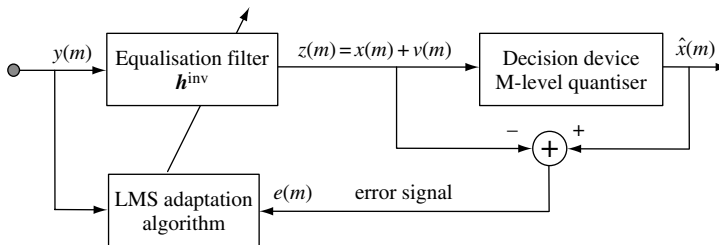


Figure 15.13 Configuration of an adaptive channel equaliser with an estimate of the channel input used as an ‘internally’ generated desired signal.

that $\hat{x}(m) = a \cdot \text{sgn}[z(m)]$, where the function $\text{sgn}(\cdot)$ gives the sign of the argument. In the following, we use a Bayesian framework to formulate the nonlinear estimator, $\psi(\cdot)$.

Assuming that the channel input is an uncorrelated process and the equaliser removes a large part of the channel distortion, the equaliser output can be expressed as the sum of the desired signal (the channel input) plus an uncorrelated additive noise term:

$$z(m) = x(m) + v(m) \quad (15.93)$$

where $v(m)$ is the so-called convolutional noise defined as

$$\begin{aligned} v(m) &= x(m) - \sum_k \hat{h}_k^{\text{inv}} y(m-k) \\ &= \sum_k (h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}) y(m-k) \end{aligned} \quad (15.94)$$

In the following, we assume that the nonlinear estimates of the channel input are correct, and hence the error signals $e(m)$ and $v(m)$ are identical. Owing to the averaging effect of the channel and the equalizer, each sample of convolutional noise is affected by many samples of the input process. From the central limit theorem, the convolutional noise, $e(m)$, can be modelled by a zero-mean Gaussian process as

$$f_E[e(m)] = \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{e^2(m)}{2\sigma_e^2}\right) \quad (15.95)$$

where σ_e^2 , the noise variance, can be estimated using the recursive time-update equation

$$\sigma_e^2(m) = \rho\sigma_e^2(m-1) + (1-\rho)e^2(m) \quad (15.96)$$

where $\rho < 1$ is the adaptation factor. The Bayesian estimate of the channel input given the equaliser output can be expressed in a general form as

$$\hat{x}(m) = \arg \min_{\hat{x}(m)} \int_C [x(m), \hat{x}(m)] f_{X|Z}[x(m)|z(m)] dx(m) \quad (15.97)$$

where $C[x(m), \hat{x}(m)]$ is a cost function and $f_{X|Z}[x(m)|z(m)]$ is the posterior pdf of the channel input signal. The choice of the cost function determines the type of the estimator, as described in Chapter 4. Using a uniform cost function in Equation (15.97) yields the MAP estimate:

$$\begin{aligned} \hat{x}^{\text{MAP}}(m) &= \arg \max_{x(m)} f_{X|Z}[x(m)|z(m)] \\ &= \arg \max_{x(m)} f_E[z(m) - x(m)] P_X[x(m)] \end{aligned} \quad (15.98)$$

Now, as an example consider an M -ary pulse amplitude modulation system, and let $\{a_i \ i = 1, \dots, M\}$ denote the set of M pulse amplitudes with a probability mass function

$$P_X[x(m)] = \sum_{i=1}^M P_i \delta[x(m) - a_i] \quad (15.99)$$

The pdf of the equaliser output, $z(m)$, can be expressed as the mixture pdf:

$$f_Z[z(m)] = \sum_{i=1}^M P_i f_E[x(m) - a_i] \quad (15.100)$$

The posterior density of the channel input is

$$P_{X|Z}[x(m) = a_i | z(m)] = \frac{1}{f_Z[z(m)]} f_E[z(m) - a_i] P_X[x(m) = a_i] \quad (15.101)$$

and the MAP estimate is obtained from

$$\hat{x}^{\text{MAP}}(m) = \arg \max_{a_i} \{f_E[z(m) - a_i] P_X[x(m) = a_i]\} \quad (15.102)$$

Note that the classification of the continuous-valued equaliser output, $z(m)$, into one of M discrete channel input symbols is basically a nonlinear process. Substitution of the zero-mean Gaussian model for the convolutional noise, $e(m)$, in Equation (102) yields

$$\hat{x}^{\text{MAP}}(m) = \arg \max_{a_i} \left(P_X[x(m) = a_i] \exp \left\{ -\frac{[z(m) - a_i]^2}{2\sigma_e^2} \right\} \right) \quad (15.103)$$

Note that, when the symbols are equiprobable, the MAP estimate reduces to a simple threshold decision device. Figure 15.13 shows a channel equaliser followed by an M -level quantiser. In this system, the output of the equaliser filter is passed to an M -ary decision circuit. The decision device, which is essentially an M -level quantiser, classifies the channel output into one of M valid symbols. The output of the decision device is taken as an internally generated desired signal to direct the equaliser adaptation.

15.5.2 EQUALISATION OF A BINARY DIGITAL CHANNEL

Consider a binary PAM communication system with an input symbol alphabet $\{a_0, a_1\}$ and symbol probabilities $P(a_0) = P_0$ and $P(a_1) = P_1 = 1 - P_0$. The pmf of the amplitude of the channel input signal can be expressed as

$$P[x(m)] = P_0 \delta[x(m) - a_0] + P_1 \delta[x(m) - a_1] \quad (15.104)$$

Assume that, at the output of the linear adaptive equaliser in Figure 15.13, the convolutional noise $v(m)$ is a zero-mean Gaussian process with variance σ_v^2 . Therefore the pdf of the equalizer output, $z(m) = x(m) + v(m)$, is a mixture of two Gaussian pdfs and can be described as

$$f_Z[z(m)] = \frac{P_0}{\sqrt{2\pi}\sigma_v} \exp \left\{ -\frac{[z(m) - a_0]^2}{2\sigma_v^2} \right\} + \frac{P_1}{\sqrt{2\pi}\sigma_v} \exp \left\{ -\frac{[z(m) - a_1]^2}{2\sigma_v^2} \right\} \quad (15.105)$$

The MAP estimate of the channel input signal is

$$\hat{x}(m) = \begin{cases} a_0 & \text{if } \frac{P_0}{\sqrt{2\pi}\sigma_v} \exp \left\{ -\frac{[z(m) - a_0]^2}{2\sigma_v^2} \right\} < \frac{P_1}{\sqrt{2\pi}\sigma_v} \exp \left\{ -\frac{[z(m) - a_1]^2}{2\sigma_v^2} \right\} \\ a_1 & \text{otherwise} \end{cases} \quad (15.106)$$

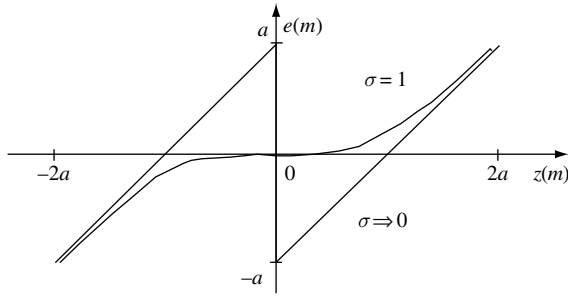


Figure 15.14 Comparison of the error functions produced by the hard nonlinearity of a signum function, Equation (15.107), and the soft nonlinearity of Equation (15.108).

For the case when the channel alphabet consists of $a_0 = -a$, $a_1 = a$ and $P_0 = P_1$, the MAP estimator is identical to the signum function $\text{sgn}[x(m)]$, and the error signal is given by

$$e(m) = z(m) - \text{sgn}[z(m)]a \quad (15.107)$$

Figure 15.14 shows the error signal as a function of $z(m)$. An undesirable property of a hard nonlinearity, such as the $\text{sgn}(\cdot)$ function is that it produces a large error signal at those instances when $z(m)$ is around zero, and a decision based on the sign of $z(m)$ is most likely to be incorrect.

A large error signal based on an incorrect decision would have an unsettling effect on the convergence of the adaptive equaliser. It is desirable to have an error function that produces small error signals when $z(m)$ is around zero. Nowlan and Hinton proposed a soft nonlinearity of the following form

$$e(m) = z(m) - \frac{e^{2az(m)/\sigma^2} - 1}{e^{2az(m)/\sigma^2} + 1}a \quad (15.108)$$

The error, $e(m)$, is small when the magnitude of $z(m)$ is small and large when magnitude of $z(m)$ is large.

15.6 EQUALISATION BASED ON HIGHER-ORDER STATISTICS

The second-order statistics of a random process, namely the autocorrelation or its Fourier transform the power spectrum, are central to the development the linear estimation theory, and form the basis of most statistical signal processing methods, such as Wiener filters and linear predictive models. An attraction of the correlation function is that a Gaussian process, of a known mean vector, can be completely described in terms of the covariance matrix, and many random processes can be well characterised by Gaussian or mixture Gaussian models. A shortcoming of second-order statistics is that they do not include the phase characteristics of the process. Therefore, given the channel output, it is not possible to estimate the channel phase from the second-order statistics. Furthermore, as a Gaussian process of known mean depends entirely on the autocovariance function, it follows that blind deconvolution, based on a Gaussian model of the channel input, cannot estimate the channel phase.

Higher-order statistics, and the probability models based on them, can model both the magnitude and the phase characteristics of a random process. In this section, we consider blind deconvolution based on higher-order statistics and their Fourier transforms, known as the higher-order spectra. The prime motivation in using higher-order statistics is their ability to model the phase characteristics. Further motivations are their potential to model channel nonlinearities, and to estimate a non-Gaussian signal in a high level of Gaussian noise.

15.6.1 HIGHER-ORDER MOMENTS, CUMULANTS AND SPECTRA

The k th order moment of a random variable X is defined as

$$\begin{aligned}
 m_k &= \mathcal{E}[x^k] \\
 &= (-j)^k \frac{\partial^k \Phi_X(\omega)}{\partial \omega^k} \Big|_{\omega=0}
 \end{aligned}
 \tag{15.109}$$

where $\Phi_X(\omega)$ is the characteristic function of the random variable X defined as

$$\Phi_X(\omega) = \mathcal{E}[\exp(j\omega x)]
 \tag{15.110}$$

From Equations (15.109) and (15.110), the first moment of X is $m_1 = \mathcal{E}[x]$, the second moment of X is $m_2 = \mathcal{E}[x^2]$, and so on. The joint k th-order moment ($k = k_1 + k_2$) of two random variables X_1 and X_2 is defined as

$$\mathcal{E}[x_1^{k_1} x_2^{k_2}] = (-j)^{k_1+k_2} \frac{\partial^{k_1} \partial^{k_2} \Phi_{X_1 X_2}(\omega_1, \omega_2)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2}} \Big|_{\omega_1=\omega_2=0}
 \tag{15.111}$$

and in general the joint k th order moment of N random variables is defined as

$$\begin{aligned}
 m_k &= \mathcal{E}[x_1^{k_1} x_2^{k_2} \dots x_N^{k_N}] \\
 &= (-j)^k \frac{\partial^k \Phi(\omega_1, \omega_2, \dots, \omega_N)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \dots \partial \omega_N^{k_N}} \Big|_{\omega_1=\omega_2=\dots=\omega_N=0}
 \end{aligned}
 \tag{15.112}$$

where $k = k_1 + k_2 + \dots + k_N$ and the joint characteristic function is

$$\Phi(\omega_1, \omega_2, \dots, \omega_N) = \mathcal{E}[\exp(j\omega_1 x_1 + \omega_2 x_2 + \dots + \omega_N x_N)]
 \tag{15.113}$$

Now the higher-order moments can be applied for characterization of discrete-time random processes. The k th-order moment of a random process $x(m)$ is defined as

$$m_x(\tau_1, \tau_2, \dots, \tau_{K-1}) = \mathcal{E}[x(m), x(m + \tau_1)x(m + \tau_2) \dots x(m + \tau_{k-1})]
 \tag{15.114}$$

Note that the second-order moment $\mathcal{E}[x(m)x(m + \tau)]$ is the autocorrelation function.

15.6.1.1 Cumulants

Cumulants are similar to moments; the difference is that the moments of a random process are derived from the characteristic function $\Phi_X(\omega)$, whereas the cumulant generating function, $C_X(\omega)$, is defined as the logarithm of the characteristic function as

$$C_X(\omega) = \ln \Phi_X(\omega) = \ln \mathcal{E}[\exp(j\omega x)] \quad (15.115)$$

Using a Taylor series expansion of the term $\mathcal{E}[\exp(j\omega x)]$ in Equation (15.115), the cumulant generating function can be expanded as

$$C_X(\omega) = \ln \left(1 + m_1(j\omega) + \frac{m_2}{2!}(j\omega)^2 + \frac{m_3}{3!}(j\omega)^3 + \cdots + \frac{m_n}{n!}(j\omega)^n + \cdots \right) \quad (15.116)$$

where $m_k = \mathcal{E}[x^k]$ is the k th moment of the random variable x . The k th-order cumulant of a random variable is defined as

$$c_k = (-j)^k \left. \frac{\partial^k C_X(\omega)}{\partial \omega^k} \right|_{\omega=0} \quad (15.117)$$

From Equations (15.116) and (15.117), we have

$$c_1 = m_1 \quad (15.118)$$

$$c_2 = m_2 - m_1^2 \quad (15.119)$$

$$c_3 = m_3 - 3m_1m_2 + 2m_1^3 \quad (15.120)$$

and so on. The general form of the k th-order ($k = k_1 + k_2 + \cdots + k_N$) joint cumulant generating function is

$$c_{k_1 \cdots k_N} = (-j)^{k_1 + \cdots + k_N} \left. \frac{\partial^{k_1 + \cdots + k_N} \ln \Phi_X(\omega_1, \dots, \omega_N)}{\partial \omega_1^{k_1} \cdots \partial \omega_N^{k_N}} \right|_{\omega_1 = \omega_2 = \cdots = \omega_N = 0} \quad (15.121)$$

The cumulants of a zero mean random process $x(m)$ are given as

$$c_x = \mathcal{E}[x(k)] = m_x = 0 \quad (\text{mean}) \quad (15.122)$$

$$\begin{aligned} c_x(k) &= \mathcal{E}[x(m)x(m+k)] - \mathcal{E}[x(m)]^2 \\ &= m_x(k) - m_x^2 = m_x(k) \quad (\text{covariance}) \end{aligned} \quad (15.123)$$

$$\begin{aligned} c_x(k_1, k_2) &= m_x(k_1, k_2) - m_x[m_x(k_1) + m_x(k_2) + m_x(k_2 - k_1)] + 2(m_x)^3 \\ &= m_x(k_1, k_2) \quad (\text{skewness}) \end{aligned} \quad (15.124)$$

$$\begin{aligned} c_x(k_1, k_2, k_3) &= m_x(k_1, k_2, k_3) - m_x(k_1)m_x(k_3 - k_2) \\ &\quad - m_x(k_2)m_x(k_3 - k_1) - m_x(k_3)m_x(k_2 - k_1) \end{aligned} \quad (15.125)$$

and so on. Note that $m_x(k_1, k_2, \dots, k_N) = \mathcal{E}[x(m)x(m+k_1), x(m+k_2), \dots, x(m+k_N)]$. The general formulation of the k th-order cumulant of a random process $x(m)$ (Rosenblatt) is defined as

$$c_x(k_1, k_2, \dots, k_n) = m_x(k_1, k_2, \dots, k_n) - m_x^G(k_1, k_2, \dots, k_n) \quad \text{for } n = 3, 4, \dots \quad (15.126)$$

where $m_x^G(k_1, k_2, \dots, k_n)$ is the k th-order moment of a Gaussian process having the same mean and autocorrelation as the random process $x(m)$. From Equation (15.126), it follows that, for a Gaussian process, the cumulants of order greater than 2 are identically zero.

15.6.1.2 Higher-order Spectra

The k th-order spectrum of a signal $x(m)$ is defined as the $(k - 1)$ -dimensional Fourier transform of the k th-order cumulant sequence as

$$C_X(\omega_1, \dots, \omega_{k-1}) = \frac{1}{(2\pi)^{k-1}} \sum_{\tau_1=-\infty}^{\infty} \dots \sum_{\tau_{k-1}=-\infty}^{\infty} c_x(\tau_1, \dots, \tau_{k-1}) e^{-j(\omega_1\tau_1 + \dots + \omega_{k-1}\tau_{k-1})} \tag{15.127}$$

For the case $k = 2$, the second-order spectrum is the power spectrum, given as

$$C_X(\omega) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} c_x(\tau) e^{-j\omega\tau} \tag{15.128}$$

The *bi-spectrum* is defined as

$$C_X(\omega_1, \omega_2) = \frac{1}{(2\pi)^2} \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} c_x(\tau_1, \tau_2) e^{-j(\omega_1\tau_1 + \omega_2\tau_2)} \tag{15.129}$$

and the *tri-spectrum* is

$$C_X(\omega_1, \omega_2, \omega_3) = \frac{1}{(2\pi)^3} \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} \sum_{\tau_3=-\infty}^{\infty} c_x(\tau_1, \tau_2, \tau_3) e^{-j(\omega_1\tau_1 + \omega_2\tau_2 + \omega_3\tau_3)} \tag{15.130}$$

Since the term $e^{j\omega t}$ is periodic with a period of 2π , it follows that higher-order spectra are periodic in each ω_k with a period of 2π .

15.6.2 HIGHER-ORDER SPECTRA OF LINEAR TIME-INVARIANT SYSTEMS

Consider a linear time-invariant system with an impulse response sequence $\{h_k\}$, input signal $x(m)$ and output signal $y(m)$. The relation between the k th-order cumulant spectra of the input and output signals is given by

$$C_Y(\omega_1, \dots, \omega_{k-1}) = H(\omega_1) \dots H(\omega_{k-1}) H^*(\omega_1 + \dots + \omega_{k-1}) C_X(\omega_1, \dots, \omega_{k-1}) \tag{15.131}$$

where $H(\omega)$ is the frequency response of the linear system $\{h_k\}$. The magnitude of the k th-order spectrum of the output signal is given as

$$|C_Y(\omega_1, \dots, \omega_{k-1})| = |H(\omega_1)| \dots |H(\omega_{k-1})| |H(\omega_1 + \dots + \omega_{k-1})| |C_X(\omega_1, \dots, \omega_{k-1})| \tag{15.132}$$

and the phase of the k th-order spectrum is

$$\Phi_Y(\omega_1, \dots, \omega_{k-1}) = \Phi_H(\omega_1) + \dots + \Phi_H(\omega_{k-1}) - \Phi_H(\omega_1 + \dots + \omega_{k-1}) + \Phi_X(\omega_1, \dots, \omega_{k-1}) \tag{15.133}$$

15.6.3 BLIND EQUALISATION BASED ON HIGHER-ORDER CEPSTRA

In this section, we consider blind equalisation of a maximum-phase channel, based on higher-order cepstra. Assume that the channel can be modelled by an all-zero filter, and that its z -transfer function, $H(z)$, can be expressed as the product of a maximum-phase polynomial factor and a minimum-phase factor as

$$H(z) = GH_{\min}(z)H_{\max}(z^{-1})z^{-D} \quad (15.134)$$

$$H_{\min}(z) = \prod_{i=1}^{P_1} (1 - \alpha_i z^{-1}), \quad |\alpha_i| < 1 \quad (15.135)$$

$$H_{\max}(z^{-1}) = \prod_{i=1}^{P_2} (1 - \beta_i z), \quad |\beta_i| < 1 \quad (15.136)$$

where G is a gain factor, $H_{\min}(z)$ is a minimum-phase polynomial with all its zeros inside the unit circle, $H_{\max}(z^{-1})$ is a maximum-phase polynomial with all its zeros outside the unit circle and z^{-D} inserts D unit delays in order to make Equation (15.134) causal. The complex cepstrum of $H(z)$ is defined as

$$h_c(m) = Z^{-1} [\ln H(z)] \quad (15.137)$$

where Z^{-1} denotes the inverse z -transform. At $z = e^{j\omega}$, the z -transform is the discrete Fourier transform, and the cepstrum of a signal is obtained by taking the inverse DFT of the logarithm of the signal spectrum. In the following we consider cepstra based on the power spectrum and the higher-order spectra, and show that the higher-order cepstra have the ability to retain maximum-phase information. Assuming that the channel input, $x(m)$, is a zero-mean uncorrelated process with variance σ_x^2 , the power spectrum of the channel output can be expressed as

$$P_Y(\omega) = \frac{\sigma_x^2}{2\pi} H(\omega)H^*(\omega) \quad (15.138)$$

The cepstrum of the power spectrum of $y(m)$ is defined as

$$\begin{aligned} y_c(m) &= \text{IDFT} [\ln P_Y(\omega)] \\ &= \text{IDFT} \left[\ln \left(\frac{\sigma_x^2 G^2}{2\pi} \right) + \ln H_{\min}(\omega) + H_{\max}(-\omega) \right. \\ &\quad \left. + \ln H_{\min}^*(\omega) + H_{\max}^*(-\omega) \right] \end{aligned} \quad (15.139)$$

where IDFT is the inverse discrete Fourier transform. Substituting Equations (15.135) and (15.136) in Equation (15.139), the cepstrum can be expressed as

$$y_c(m) = \begin{cases} \ln \left(G^2 \sigma_x^2 / 2\pi \right), & m = 0 \\ - \left(A^{(m)} + B^{(m)} \right) / m, & m > 0 \\ \left(A^{(-m)} + B^{(-m)} \right) / m, & m < 0 \end{cases} \quad (15.140)$$

where $A^{(m)}$ and $B^{(m)}$ are defined as

$$A^{(m)} = \sum_{i=1}^{P_1} \alpha_i^m \tag{15.141}$$

$$B^{(m)} = \sum_{i=1}^{P_2} \beta_i^m \tag{15.142}$$

Note from Equation (15.140) that, along the index m , the maximum-phase information, $B^{(m)}$, and the minimum-phase information, $A^{(m)}$, overlap and cannot be separated.

15.6.3.1 Bi-cepstrum

The bi-cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the bi-spectrum:

$$y_c(m_1, m_2) = \text{IDFT}_2[\log C_Y(\omega_1, \omega_2)] \tag{15.143}$$

where $\text{IDFT}_2[.]$ denotes the two-dimensional inverse discrete Fourier transform. The relationship between the bi-spectra of the input and output of a linear system is

$$C_Y(\omega_1, \omega_2) = H(\omega_1)H(\omega_2)H^*(\omega_1 + \omega_2)C_X(\omega_1, \omega_2) \tag{15.144}$$

Assuming that the input, $x(m)$, of the linear time-invariant system, $\{h_k\}$, is an uncorrelated non-Gaussian process, the bi-spectrum of the output can be written as

$$C_Y(\omega_1, \omega_2) = \frac{\gamma_x^{(3)} G^3}{(2\pi)^2} H_{\min}(\omega_1)H_{\max}(-\omega_1)H_{\min}(\omega_2)H_{\max}(-\omega_2) \tag{15.145}$$

$$\times H_{\min}^*(\omega_1 + \omega_2)H_{\max}^*(-\omega_1 - \omega_2)$$

where $\gamma_x^{(3)}/(2\pi)^2$ is the third-order cumulant of the uncorrelated random input process, $x(m)$. Taking the logarithm of Equation (15.145) yields

$$\ln C_Y(\omega_1, \omega_2) = \ln |A| + \ln H_{\min}(\omega_1) + \ln H_{\max}(-\omega_1) + \ln H_{\min}(\omega_2) + \ln H_{\max}(-\omega_2) \tag{15.146}$$

$$+ \ln H_{\min}^*(\omega_1 + \omega_2) + \ln H_{\max}^*(-\omega_1 - \omega_2)$$

where $A = \gamma_x^{(3)} G^3 / (2\pi)^2$. The bi-cepstrum is obtained through the inverse DFT of Equation (15.146) as

$$y_c(m_1, m_2) = \begin{cases} \ln |A|, & m_1 = m_2 = 0 \\ -A^{(m_1)}/m_1, & m_1 > 0, m_2 = 0 \\ -A^{(m_2)}/m_2, & m_2 > 0, m_1 = 0 \\ -B^{(-m_1)}/m_1, & m_1 < 0, m_2 = 0 \\ B^{(-m_2)}/m_2, & m_2 < 0, m_1 = 0 \\ -B^{(m_2)}/m_2, & m_1 = m_2 > 0 \\ A^{(-m_2)}/m_2, & m_1 = m_2 < 0 \\ 0, & \text{otherwise} \end{cases} \tag{15.147}$$

Note from Equation (15.147) that the maximum-phase information, $B^{(m)}$, and the minimum-phase information, $A^{(m)}$, are separated and appear in different regions of the bi-cepstrum indices, m_1 and m_2 .

The higher-order cepstral coefficients can be obtained either from the IDFT of higher-order spectra, as in Equation (15.147), or using parametric methods, as follows. In general, the cepstral and cumulant coefficients can be related by a convolutional equation. Pan and Nikias (1988) have shown that the recursive relation between the bi-cepstrum coefficients and the third-order cumulants of a random process is

$$y_c(m_1, m_2) * [-m_1 c_y(m_1, m_2)] = -m_1 c_y(m_1, m_2) \quad (15.148)$$

Substituting Equation (15.147) in Equation (15.148) yields

$$\begin{aligned} \sum_{i=1}^{\infty} A^{(i)} [c_x(m_1 - i, m_2) - c_x(m_1 + i, m_2 + i)] + B^{(i)} [c_x(m_1 - i, m_2 - i) - c_x(m_1 + i, m_2)] \\ = -m_1 c_x(m_1, m_2) \end{aligned} \quad (15.149)$$

The truncation of the infinite summation in Equation (15.149) provides an approximate equation as

$$\begin{aligned} \sum_{i=1}^P A^{(i)} [c_x(m_1 - i, m_2) - c_x(m_1 + i, m_2 + i)] \\ + \sum_{i=1}^Q B^{(i)} [c_x(m_1 - i, m_2 - i) - c_x(m_1 + i, m_2)] \approx -m_1 c_x(m_1, m_2) \end{aligned} \quad (15.150)$$

Equation (15.150) can be used to solve for the cepstral parameters $A^{(m)}$ and $B^{(m)}$.

15.6.3.2 Tri-cepstrum

The tri-cepstrum of a signal $y(m)$ is defined as the inverse Fourier transform of the tri-spectrum:

$$y_c(m_1, m_2, m_3) = \text{IDFT}_3[\ln C_Y(\omega_1, \omega_2, \omega_3)] \quad (15.151)$$

where $\text{IDFT}_3[\cdot]$ denotes the three-dimensional inverse discrete Fourier transform. The tri-spectra of the input and output of the linear system are related as

$$C_Y(\omega_1, \omega_2, \omega_3) = H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1 + \omega_2 + \omega_3)C_X(\omega_1, \omega_2, \omega_3) \quad (15.152)$$

Assuming that the channel input $x(m)$ is uncorrelated, Equation (15.152) becomes

$$C_Y(\omega_1, \omega_2, \omega_3) = \frac{\gamma_x^{(4)} G^4}{(2\pi)^3} H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1 + \omega_2 + \omega_3) \quad (15.153)$$

where $\gamma_x^{(4)}/(2\pi)^3$ is the fourth-order cumulant of the input signal. Taking the logarithm of the tri-spectrum gives

$$\begin{aligned} \ln C_Y(\omega_1, \omega_2, \omega_3) = & \frac{\gamma_x^{(4)} G^4}{(2\pi)^3} + \ln H_{\min}(\omega_1) + \ln H_{\max}(-\omega_1) + \ln H_{\min}(\omega_2) + \ln H_{\max}(-\omega_2) \\ & + \ln H_{\min}(\omega_3) + \ln H_{\max}(-\omega_3) + \ln H_{\min}^*(\omega_1 + \omega_2 + \omega_3) + \ln H_{\max}^*(-\omega_1 - \omega_2 - \omega_3) \end{aligned} \quad (15.154)$$

From Equations (15.151) and (15.154), we have

$$y_c(m_1, m_2, m_3) = \begin{cases} \ln A, & m_1 = m_2 = m_3 = 0 \\ -A^{(m_1)}/m_1, & m_1 > 0, m_2 = m_3 = 0 \\ -A^{(m_2)}/m_2, & m_2 > 0, m_1 = m_3 = 0 \\ -A^{(m_3)}/m_3, & m_3 > 0, m_1 = m_2 = 0 \\ B^{(-m_1)}/m_1, & m_1 < 0, m_2 = m_3 = 0 \\ B^{(-m_2)}/m_2, & m_2 < 0, m_1 = m_3 = 0 \\ B^{(-m_3)}/m_3, & m_3 < 0, m_1 = m_2 = 0 \\ -B^{(m_2)}/m_2, & m_1 = m_2 = m_3 > 0 \\ A^{(m_2)}/m_2, & m_1 = m_2 = m_3 < 0 \\ 0 & \text{otherwise} \end{cases} \quad (15.155)$$

where $A = \gamma_x^{(4)} G^4 / (2\pi)^3$. Note from Equation (15.155) that the maximum-phase information, $B^{(m)}$, and the minimum-phase information, $A^{(m)}$, are separated and appear in different regions of the tri-cepstrum indices, m_1, m_2 and m_3 .

15.6.3.3 Calculation of Equaliser Coefficients from the Tri-cepstrum

Assuming that the channel z -transfer function can be described by Equation (15.134), the inverse channel can be written as

$$H^{\text{inv}}(z) = \frac{1}{H(z)} = \frac{1}{H_{\min}(z)H_{\max}(z^{-1})} = H_{\min}^{\text{inv}}(z)H_{\max}^{\text{inv}}(z^{-1}) \quad (15.156)$$

where it is assumed that the channel gain G is unity. In the time domain Equation (15.156) becomes

$$h^{\text{inv}}(m) = h_{\min}^{\text{inv}}(m) * h_{\max}^{\text{inv}}(m) \quad (15.157)$$

Pan and Nikias (1988) describe an iterative algorithm for estimation of the truncated impulse response of the maximum-phase and the minimum-phase factors of the inverse channel transfer function. Let $\hat{h}_{\min}^{\text{inv}}(i, m), \hat{h}_{\max}^{\text{inv}}(i, m)$ denote the estimates of the m th coefficients of the maximum-phase and minimum-phase parts of the inverse channel at the i th iteration. The Pan and Nikias algorithm is the following:

(1) initialisation

$$\hat{h}_{\min}^{\text{inv}}(i, 0) = \hat{h}_{\max}^{\text{inv}}(i, 0) = 1 \quad (15.158)$$

(2) calculation of the minimum-phase polynomial

$$\hat{h}_{\min}^{\text{inv}}(i, m) = \frac{1}{m} \sum_{k=2}^{m+1} \hat{A}^{(k-1)} \hat{h}_{\min}^{\text{inv}}(i, m-k+1) \quad i = 1, \dots, P_1 \quad (15.159)$$

(3) calculation of the maximum-phase polynomial

$$\hat{h}_{\max}^{\text{inv}}(i, m) = \frac{1}{m} \sum_{k=m+1}^0 \hat{B}^{(1-k)} \hat{h}_{\max}^{\text{inv}}(i, m-k+1) \quad i = -1, \dots, -P_2 \quad (15.160)$$

The maximum-phase and minimum-phase components of the inverse channel response are combined in Equation (15.157) to give the inverse channel equaliser.

15.7 SUMMARY

In this chapter, we considered a number of different approaches to channel equalisation. The chapter began with an introduction to models for channel distortions, the definition of an ideal channel equaliser, and the problems that arise in channel equalisation due to noise and possible noninvertibility of the channel. In some problems, such as speech recognition or restoration of distorted audio signals, we are mainly interested in restoring the magnitude spectrum of the signal, and phase restoration is not a primary objective. In other applications, such as digital telecommunications, the restoration of both the amplitude and the timing of the transmitted symbols is of interest, and hence we need to equalise for both the magnitude and the phase distortions.

In Section 15.1, we considered the least square error Wiener equaliser. The Wiener equaliser can only be used if we have access to the channel input or the cross-correlation of the channel input and output signals.

For cases where a training signal cannot be employed to identify the channel response, the channel input is recovered through a blind equalisation method. Blind equalisation is feasible only if some statistics of the channel input signal are available. In Section 15.2, we considered blind equalisation using the power spectrum of the input signal. This method was introduced by Stockham for restoration of the magnitude spectrum of distorted acoustic recordings. In Section 15.3, we considered a blind deconvolution method based on the factorisation of a linear predictive model of the convolved signals.

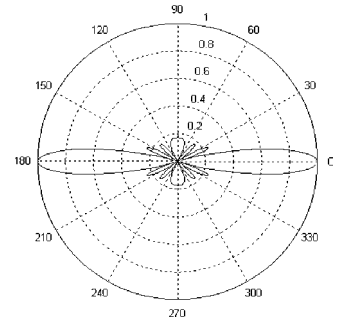
Bayesian inference provides a framework for inclusion of the statistics of the channel input and perhaps also those of the channel environment. In Section 15.4, we considered Bayesian equalisation methods, and studied the case where the channel input is modelled by a set of hidden Markov models. Section 15.5 introduced channel equalisation methods for removal of intersymbol interference in digital telecommunications systems, and finally, in Section 15.6, we considered the use of higher-order spectra for equalisation of nonminimum-phase channels.

BIBLIOGRAPHY

- Belfiore C.A. and Park J.H. (1979) Decision feedback equalization. *Proc. IEEE*, **67**: 1143–1156.
 Bellini S. (1986) Bussgang techniques for blind equalization. *IEEE GLOBECOM Conference Record*, pp. 1634–1640.

- Bellini S. and Rocca F. (1988) Near optimal blind deconvolution. *IEEE Proceedings International Conference on Acoustics, Speech, and Signal Processing, ICASSP-88*, pp. 2236–2239.
- Benveniste A., Goursat M. and Ruget G. (1980) Robust identification of a nonminimum phase system: blind adjustment of linear equalizer in data communications. *IEEE Trans. Autom. Control*, **AC-25**: 385–399.
- Gersho A. (1969) Adaptive equalization of highly dispersive channels for data transmission. *Bell Systems Technical Journal*, **48**: 55–70.
- Godard D. N. (1974) Channel equalization using a Kallman filter for fast data transmission. *IBM J. Res. Dev.*, **18**: 267–273.
- Godard D. N. (1980) Self-recovering equalization and carrier tracking in a two-dimensional data communication system. *IEEE Trans. Commun.*, **COM-28**: 1867–1875.
- Hanson B.A. and Applebaum T.H. (1993) Subband or cepstral domain filtering for recognition of Lombard and channel-distorted speech. *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing*, pp. 79–82.
- Hariharan S. and Clark A.P. (1990) HF channel estimation using a fast transversal filter algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, **38**: 1353–1362.
- Hatzinako S.D. (1990) Blind equalization based on polyspectra. Ph.D. thesis. Northeastern University, Boston, MA.
- Hermansky H. and Morgan N. (1992) Towards handling the acoustic environment in spoken language processing. *International Conference on Spoken Language Processing, Tu, fPM.1.1.1*, pp. 85–88.
- Lucky R.W. (1965a) Automatic equalization of digital communications. *Bell Systems Technical Journal*, **44**: pp. 547–588.
- Lucky R.W. (1965b) Techniques for adaptive equalization of digital communication systems. *Bell Systems Technical Journal*, **45**: 255–286.
- Mendel J.M. (1990) *Maximum Likelihood Deconvolution: a Journey into Model Based Signal Processing*. Springer, New York.
- Mendel J.M. (1991) Tutorial on higher order statistics (spectra) in signal processing and system theory: theoretical results and some applications. *Proc. IEEE*, **79**: 278–305.
- Mokbel C., Monne J. and Jouvét D. (1993) On-line adaptation of a speech recogniser to variations in telephone line conditions. *Proceedings of the 3rd European Conference on Speech Communication and Technology, EuroSpeech-93*, vol. 2, pp. 1247–1250.
- Monsen P. (1971) Feedback equalization for fading dispersive channels. *IEEE Trans. Inform. Theory*, **IT-17**: 56–64.
- Nikias C.L. and Chiang H.H. (1991) Higher-order spectrum estimation via non-causal autoregressive modeling and deconvolution. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-36**: 1911–1913.
- Nowlan S.J. and Hinton G.E. (1993) A soft decision-directed algorithm for blind equalization. *IEEE Trans. Commun.*, **41**(2): 275–279.
- Pan R. and Nikias C.L. (1988) Complex cepstrum of higher order cumulants and non-minimum phase identification. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-36**: 186–205.
- Picchi G. and Prati G. (1987) Blind equalization and carrier recovery using a stop-and-go decision-directed algorithm. *IEEE Trans. Commun.*, **COM-35**: 877–887.
- Qureshi S.U. (1985) Adaptive equalization. *IEEE Proc.*, **73**(9): 1349–1387.
- Raghuveer M.R. and Nikias C.L. (1985) Bispectrum estimation: a parametric approach. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-33**(5): 35–48.
- Rosenblatt M. (1985) *Stationary Sequences and Random Fields*. Birkhauser, Boston, MA.
- Spencer P.S. and Rayner P.J.W. (1990) Separation of stationary and time-varying systems and its applications to the restoration of gramophone recordings. Ph.D. thesis. Cambridge University, Cambridge.
- Stockham T.G., Cannon T.M. and Ingebretsen R.B. (1975) Blind deconvolution through digital signal processing. *IEEE Proc.*, **63**(4): 678–692.
- Ungerboeck G. (1972) Theory on the speed of convergence in adaptive equalizers for digital communication. *IBM J. Res. Dev.*, **16**: 546–555.

16



Speech Enhancement in Noise

16.1 Introduction	410
16.2 Single-input Speech-enhancement Methods	411
16.3 Multiple-input Speech-enhancement Methods	425
16.4 Speech Distortion Measurements	430
Bibliography	431

Speech enhancement in noisy environments, such as in cars, trains, streets and at noisy public venues, improves the quality and intelligibility of speech. Noise reduction benefits a wide range of applications, such as mobile phones, hands-free phones, teleconferencing, in-car cabin communication systems and automated speech recognition services. This chapter provides an overview of the main methods for single-input and multiple-input speech enhancement in noise.

The classical noise reduction method is the Wiener filter, which modifies the magnitude frequency spectrum of the noisy input signal in proportion to an estimate of the signal-to-noise ratio at each frequency. The Wiener filter requires estimates of the power spectra, or equivalently the correlation matrices, of speech and noise. The simplest form of noise reduction is based on subtraction of an estimate of the magnitude spectrum of the noise from that of the noisy signal. Spectral subtraction only requires an estimate of the average noise spectrum, usually obtained from the speech inactive noise-only periods, but introduces some distortions. High-performance speech enhancement methods are based on Bayesian estimation and include Kalman filters, minimum mean squared error estimation and, ultimately, the signal restoration methods that are based on hidden Markov models of speech and noise. Multiple-input noise reduction methods are particularly useful for teleconferencing and in-car communication and include beam forming arrays, adaptive noise cancellation and stereophonic echo cancellation.

16.1 INTRODUCTION

De-noising speech improves the quality and the intelligibility of voice communication in noisy environments and reduces communication fatigue. Noise reduction benefits the users of hands-free phones, mobile phones and voice-controlled automated services used in noisy moving environments such as cars, trains, streets, conference halls and other public venues.

Figure 16.1 illustrates a classification of the main signal processing methods for enhancement of noisy speech into two broad types:

- (1) Single-input speech enhancement systems, where the only available signal is the noise-contaminated speech picked up by a single microphone. Single input systems do not *cancel* noise, rather they *suppress* the noise using estimates of the signal-to-noise ratio of the frequency spectrum of the input signal. Single-input systems rely on the statistical models of speech and noise, which may be estimated from the speech-inactive periods or decoded from a set of pre-trained models of speech and noise. An example of a useful application of a single-input enhancement system is a mobile phone system used in noisy environments.
- (2) Multiple-input speech enhancement systems, where a number of signals containing speech and noise are picked up by several microphones. Examples of multiple-inputs systems are adaptive noise cancellation, adaptive beam-forming microphone arrays and multiple-input multiple-output (MIMO) acoustic echo cancellation systems. In multiple-input systems the microphones can be designed, spatially arranged and adapted for optimum performance. Multiple-input noise-reduction systems are useful for teleconferencing systems and for in-car cabin communication systems.

Figure 16.1 illustrates a categorization of the main noise reduction methods used for single-input and multiple-input scenarios. In order to achieve the best noise-reduction performance, where possible, the advantages of the signal processing methods developed for single-input noise suppression and multiple-inputs noise cancellation systems are combined.

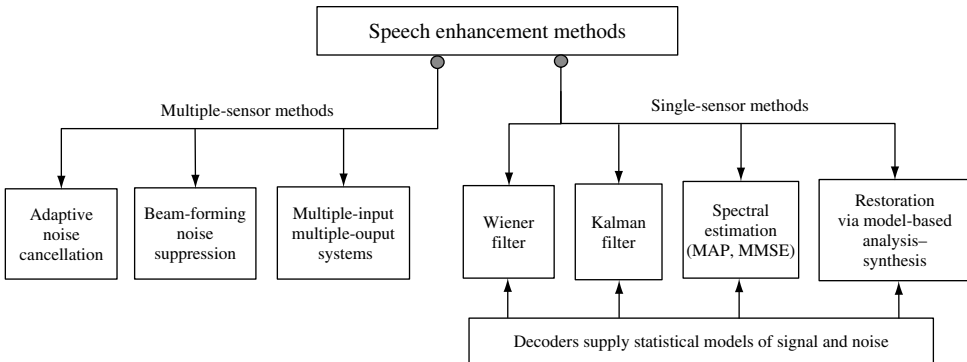


Figure 16.1 A categorization of speech enhancement methods. Note that statistical models can optionally provide single-input noise reduction methods with the additional information needed for improved performance.

16.2 SINGLE-INPUT SPEECH-ENHANCEMENT METHODS

In single-input systems the only available signal is the noisy speech; however, in applications where speech enhancement and recognition are performed on the same system, the results of speech recognition can provide the speech enhancement method with such information as the statistics of the power spectra or correlation matrices obtained from decoding the most-likely speech and noise models. Single-input noise-reduction methods include Wiener filter, spectral subtraction, Kalman filter, MMSE method and speech restoration via model-based analysis and synthesis methods, as described in this section.

16.2.1 AN OVERVIEW OF A SPEECH-ENHANCEMENT SYSTEM

Assuming that the speech signal, $x(m)$, and the noise, $n(m)$, are additive, the noisy speech, $y(m)$, is modelled as

$$y(m) = x(m) + n(m) \quad (16.1)$$

where the integer variable m denotes the discrete-time index. It is generally assumed that the speech is not correlated with noise; this is a reasonable assumption in most cases when the signal and noise are generated by independent sources.

The general form of a typical speech-enhancement method is shown in Figure 16.2. The speech-enhancement system is composed of a combination of the following modules:

- (1) speech segmentation into a sequence of overlapping frames (of about 20–30 ms) followed by windowing of each segment with a popular window such as the Hann window;
- (2) discrete Fourier transformation of the speech samples within each frame to a set of short-time spectral samples;
- (3) estimation of the spectral amplitudes of clean speech – this involves a modification of the magnitude spectrum of noisy speech according to an estimate of the signal to noise ratio at each frequency;
- (4) an inter-frame signal smoothing method to utilise the temporal correlations of the spectral values across successive frames of speech;
- (5) speech and noise models, and a speech and noise decoder, to supply the speech estimator with the required statistics (power spectra, correlation matrices, etc.) of speech and noise;
- (6) voice activity detection, used to estimate and adapt noise models from the noise-only periods and also for applying extra attenuation to noise-only periods.

In the following, the elements of a speech-enhancement system are described in more detail.

16.2.1.1 Segmentation and Windowing of Speech

Speech processing systems divide the sampled speech signal into overlapping frames of about 20–30 ms duration. The N speech samples within each frame are processed and represented by a set of spectral features or by a linear prediction model of speech production.

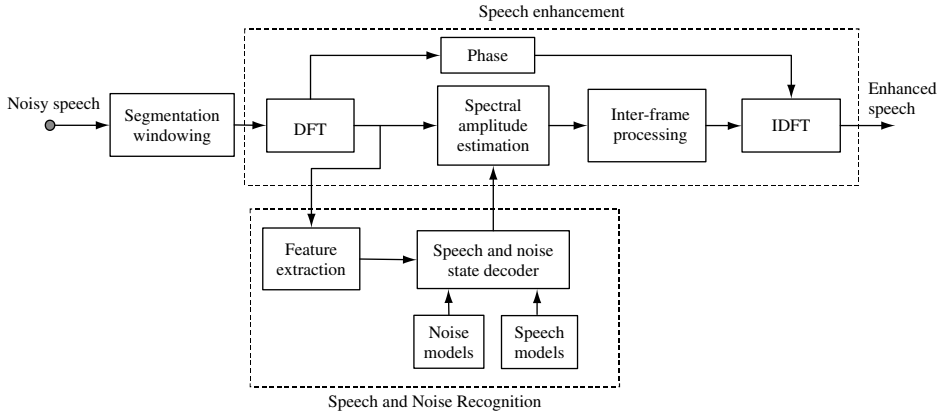


Figure 16.2 Block diagram illustration of a speech-enhancement system.

The signal within each frame is assumed to be a stationary process. The choice of the length of speech frames (typically set to between 20 and 30 ms) is constrained by the stationarity assumption of linear time-invariant signal processing methods such as Fourier transform or linear prediction model, and by the maximum allowable delay for real-time communication systems such as voice coders.

16.2.1.2 Spectral Representation of Speech and Noise

Speech is segmented into overlapping frames of N samples and transformed to frequency domain via discrete Fourier transform. In the frequency domain, the noisy speech can be represented as

$$Y(k) = X(k) + N(k) \quad k = 0, \dots, N-1 \quad (16.2)$$

where $X(k)$, $N(k)$ and $Y(k)$ are the short-time discrete Fourier transforms of speech, noise and noisy speech, respectively. The integer k represents the discrete frequency variable; it corresponds to an actual frequency of $2k\pi/N$ (rad/s) or kF_s/N (Hz) where F_s is the sampling frequency.

Equation (16.2) can be written in the complex polar form in terms of the magnitudes and the phases of the signal and noise at discrete frequency k as

$$Y_k e^{j\theta_{Y_k}} = X_k e^{j\theta_{X_k}} + N_k e^{j\theta_{N_k}} \quad k = 0, \dots, N-1 \quad (16.3)$$

where $Y_k = |Y(k)|$ and $\theta_{Y_k} = \tan^{-1} \{ \text{Im}[Y(k)] / \text{Re}[Y(k)] \}$ are the magnitude and phase of the frequency spectrum, respectively. Note that the Fourier transform models the correlation of speech samples with sinusoidal basis functions. The DFT bins can then be processed individually or in groups of frequencies, taking into account the psychoacoustics of hearing.

16.2.1.3 Linear Prediction Model Representation of Speech and Noise

The correlation of speech (or noise) samples can be modelled with a linear prediction (aka autoregressive) model, introduced in Chapter 8. Using a linear prediction model of speech and noise, the noisy speech is expressed as

$$y(m) = \underbrace{\sum_{k=1}^p a_k x(m-k)}_{\text{Speech model}} + e(m) + \underbrace{\sum_{k=1}^q b_k n(m-k)}_{\text{Noise model}} + v(m) \quad (16.4)$$

where a_k and b_k are the coefficients of linear prediction models of speech and noise, respectively. Linear prediction models can be used in a variety of speech enhancement methods, including Wiener filters, Kalman filters and speech restoration via decomposition and re-synthesis.

16.2.1.4 Interframe and Intraframe Correlations

The two main issues in modelling noisy speech are:

- (1) modelling and utilization of the probability distributions and the intraframe correlations of speech and noise samples *within* each noisy speech frame;
- (2) modelling and utilization of the probability distributions and the interframe correlations of speech and noise features *across* successive frames of noisy speech.

Most speech-enhancement systems are based on estimates of the short-time amplitude spectrum or the linear prediction model of speech. The phase distortion of speech is ignored. In the case of DFT-based features, each spectral sample, $X(k)$, at a discrete frequency k is the correlation of speech samples, $x(m)$, with a sinusoidal basis function $e^{-j2\pi km/N}$. The intraframe spectral correlation, that is the correlation of spectral samples within a frame of speech, is often ignored, as is the inter-frame temporal correlation of spectral samples across successive speech frames.

In the case of linear prediction models, the poles model the spectral correlations within each frame. However, the denoising of linear prediction model poles, or coefficients, is achieved through denoising the frequency response of clean speech and that ignores the correlation of spectral samples. The optimal utilization of the interframe and intraframe correlations of speech samples is a continuing research issue.

16.2.1.5 Speech Estimation Module

At the heart of a speech-enhancement system is the speech-estimation module. For speech enhancement, usually the spectral amplitude, or a linear prediction model, of speech is estimated and this estimate is subsequently used to reconstruct speech samples.

A variety of methods have been proposed for the estimation of clean speech, including the Wiener filter, spectral subtraction, Kalman filters, the minimum mean squared error and the maximum *a posteriori* method. For proper functioning of the speech estimation module a knowledge of the statistics of speech and noise is required and this can be estimated from the noisy speech or it can be obtained from pre-trained models of speech and noise.

16.2.1.6 Probability Models of Speech and Noise

The implementation of a noise-reduction method such as the Wiener filter, Kalman filter, spectral subtraction or a Bayesian estimation method requires estimates of the statistics (and in particular the power spectra or equivalently the correlation matrices) of the speech and noise. An estimate of the noise statistics can be obtained from speech-inactive periods; however, for best results the speech and noise statistics are obtained from a network of probability models of speech and noise, and this essentially implies that in an optimal speech processing system speech recognition and speech enhancement would need to be integrated.

The most commonly used probability models for speech are hidden Markov models. Hidden Markov models, or alternatively Gaussian mixture models, can also be used for modelling nonstationary noise. To model different types of noise, a number of HMMs need to be trained, one HMM for each type of noise. Alternatively, one can use a GMM of noise with a large number of components, with each component effectively modelling a different type of noise.

16.2.2 WIENER FILTER FOR DE-NOISING SPEECH

The Wiener filter theory, introduced in Chapter 6, forms the foundation of speech de-noising systems. The output of a Wiener filter is given by

$$\hat{x}(m) = \sum_{i=0}^P w(i)y(m-i) \quad (16.5)$$

where $w(k)$ is the filter coefficient for de-noising the input speech $y(m)$ and $\hat{x}(m)$ is the estimate of clean speech $x(m)$. The Wiener filter coefficient vector, $\mathbf{w} = [w(0), w(1), \dots, w(P)]^T$, was derived in Chapter 6 as

$$\mathbf{w} = \mathbf{R}_{yy}^{-1} \mathbf{r}_{xy} \quad (16.6)$$

where \mathbf{R}_{yy} is the autocorrelation matrix of the noisy speech signal, \mathbf{y} , and \mathbf{r}_{xy} is the cross correlation vector of the clean speech, \mathbf{x} , and noisy speech, \mathbf{y} .

For uncorrelated speech and noise, the Wiener filter Equation (16.6) can be written as

$$\mathbf{w} = [\mathbf{R}_{xx} + \mathbf{R}_{nn}]^{-1} \mathbf{r}_{xx} \quad (16.7)$$

where \mathbf{R}_{xx} and \mathbf{R}_{nn} are the autocorrelation matrices of the speech and noise, respectively, and \mathbf{r}_{xx} is the autocorrelation vector of the speech. In the frequency domain, for additive noise uncorrelated with speech, the Wiener filter equation was derived in Chapter 6 as

$$W(k) = \frac{P_{XX}(k)}{P_{XX}(k) + P_{NN}(k)} \quad (16.8)$$

where $W(k)$ is the frequency response of the Wiener filter, $P_{XX}(k)$ and $P_{NN}(k)$ are the power spectra of speech and noise, respectively, and k is the discrete frequency variable. Figure 16.3 outlines a block diagram implementation of a frequency domain Wiener filter.

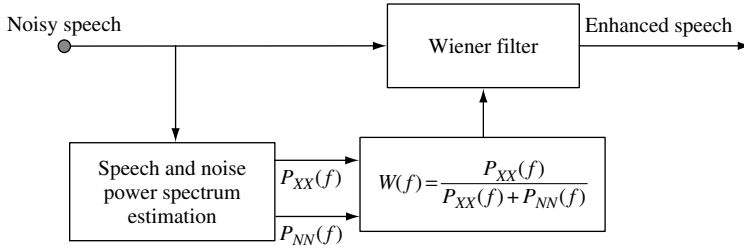


Figure 16.3 Block diagram overview of implementation of a Wiener filter for a speech-enhancement system.

By dividing the numerator and the denominator of Equation (16.8) by $P_{NN}(k)$, the Wiener filter can be expressed in terms of the signal-to-noise ratio as

$$W(k) = \frac{\text{SNR}(k)}{\text{SNR}(k) + 1} \quad (16.9)$$

This equation reveals an important aspect of the general workings of the signal-input noise reduction system: *noise suppression methods effectively use a function of the estimates of the signal-to-noise ratios to modify the spectral amplitudes of the noisy signal.*

16.2.2.1 Wiener Filter Based on Linear Prediction Models

Wiener filters employing linear prediction models of speech and noise may be used for speech enhancement. The frequency response of Wiener filter can be expressed in terms of the ratio of power spectra of autoregressive (i.e. linear prediction) models of speech and noise as

$$W(f) = \frac{P_{XX}(f)}{P_{YY}(f)} = \frac{G_X^2/A_X^2(f)}{G_Y^2/A_Y^2(f)} = \frac{G_X^2}{G_Y^2} \frac{A_Y^2(f)}{A_X^2(f)} \quad (16.10)$$

where $G_X(f)/A_X(f)$ and $G_Y(f)/A_Y(f)$ are the frequency responses of linear prediction models of speech and noisy speech, respectively. In the time domain a square root version of Wiener filter equation (16.10) can be implemented as

$$\hat{x}(m) = \sum_{k=1}^p a_x(k) \hat{x}(m-k) + \frac{G_X}{G_Y} \sum_{k=0}^q a_y(k) y(m-k) \quad (16.11)$$

where $a_x(k)$ and $a_y(k)$ are the coefficients of autoregressive models of clean speech and noisy speech, respectively.

16.2.2.2 HMM-based Wiener Filters

The key to the successful implementation of a Wiener filter is the accurate estimation of the power spectra of speech and noise, $P_{XX}(k)$ and $P_{NN}(k)$, for the frequency domain Wiener

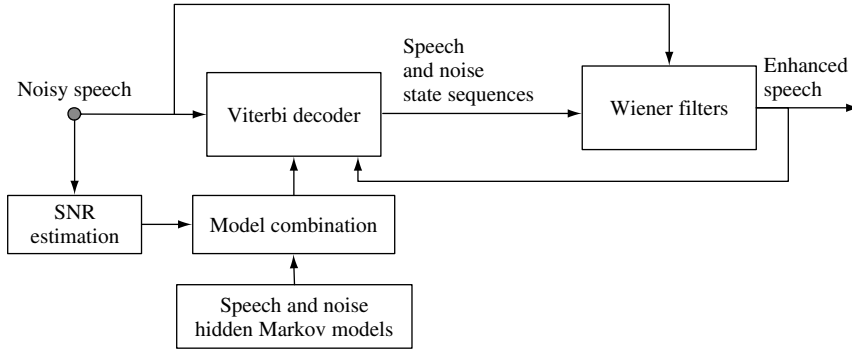


Figure 16.4 Block diagram illustration of a speech-enhancement system based on Wiener filters and hidden Markov models.

filter of Equation (16.8) or equivalently the estimation of the correlation matrices of speech and noise, \mathbf{R}_{xx} and \mathbf{R}_{nn} , for the time domain Wiener filter of Equation (16.7). This is not a trivial task as speech and most noise processes are nonstationary.

Given the noisy speech signal, the time-varying power spectra of speech and noise may be estimated from a set of pre-trained hidden Markov models, or Gaussian mixture models, of speech and noise using a Viterbi decoder (Figure 16.4). An HMM state-based Wiener filter involves the following signal processing steps:

- (1) Speech and noise decomposition – this involves the estimation of the most likely combination of speech and noise HMMs given the noisy speech signal. Using Viterbi state decoders, the most likely combination of speech and noise states yield the pdfs of the spectra of the most likely estimates of the clean speech and noise.
- (2) The speech and noise power spectra from (1) are used to implement state-based Wiener filters.

In HMM-based Wiener filtering, the choice of the speech features for training HMMs needs to be appropriate for both speech recognition and enhancement. Linear prediction-based cepstrum features provide a suitable choice as the cepstrum coefficients obtained from HMM states can be mapped to the linear prediction model coefficients and thereafter to the linear prediction model spectrum for use in the implementation of the Wiener filter.

Assuming that for a noisy speech signal spectrum, $Y(k)$, the Viterbi decoder returns M different most likely state sequences, and that in each state the probability density function of the speech spectrum is represented by a mixture of L Gaussian pdfs, the Wiener filter is given by

$$\hat{X}(k) = \left[\sum_{\beta=1}^M \sum_{\gamma=1}^L p(\beta, \gamma) W_{\beta, \gamma}(k) \right] Y(k) \quad (16.12)$$

where $p(\beta, \gamma)$ is the estimated probability of speech and noise spectra from mixture γ of HMM state β and $W_{\beta, \gamma}(K)$ is the state Wiener filter.

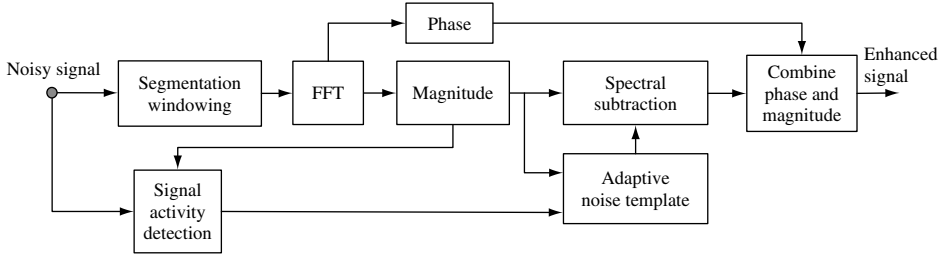


Figure 16.5 Block diagram illustration of an FFT-based spectral subtraction system for de-noising speech.

16.2.3 SPECTRAL SUBTRACTION OF NOISE

A simple and widely studied speech enhancement method is the spectral subtraction method described in Chapter 11 (Figure 16.5). In spectral subtraction an estimate of the average magnitude spectrum of the noise is subtracted from the magnitude spectrum of noisy speech. As explained in Chapter 11, the spectral subtraction filter can be expressed as the product of the noisy speech spectrum, $Y(k)$, and a spectral gain function, $W_{SS}(k)$

$$\hat{X}(k) = W_{SS}(k)Y(k) \quad (16.13)$$

where the frequency response of the spectral subtraction filter, $W_{SS}(k)$, is

$$W_{SS}(k) = \text{fn} \left[1 - \frac{\alpha(k)\hat{N}(k)}{Y(k)} \right] \quad (16.14)$$

where $\hat{N}(k)$ is an estimate of the noise average amplitude spectrum, $\alpha(k)$ is a frequency-dependent subtraction factor and the function $\text{fn}(\cdot)$ is chosen to avoid negative values of $W_{SS}(k)$ and provide a smoother frequency response when the signal-to-noise ratio drops to relatively lower values. The form of the function $\text{fn}(\cdot)$ can be chosen as

$$W_{SS}(k) = \begin{cases} 1 - \alpha(k)\hat{N}(k)/Y(k) & \text{if } \text{SNR}(k) < \text{SNR}_{\text{Thresh}} \\ \gamma \exp \{-\beta [\text{SNR}_{\text{Thresh}} - \text{SNR}(k)]\} & \text{else} \end{cases} \quad (16.15)$$

where $\text{SNR}(k)$ is an estimate of the signal-to-noise ratio at the discrete frequency k and $\text{SNR}_{\text{Thresh}}$ is a threshold SNR below which spectral subtraction switches to a form of exponential attenuation, γ is a parameter that provides continuity at the switching point and β is an attenuation control factor.

The problem with spectral subtraction is that it often distorts the speech and results in the appearance of annoying short bursts of noise. The shortcomings of spectral subtraction method can be summarized as follows:

- (1) The only statistics used in spectral subtraction is the mean of the magnitude spectrum of the noise. The mean and variance of the clean speech and the variance of the noise are not employed in the estimation process. Consequently noise variations about the mean

are not suppressed and this results in more distortions than would be the case if the variance information were also used.

- (2) A hard decision needs to be employed to avoid the values of the estimates of the magnitude spectrum after subtraction going negative or below a noise floor value.
- (3) The spectral subtraction method is not speech-specific; the spectral trajectories of speech across time are not modelled and used in the de-noising process.

16.2.3.1 Spectral Subtraction Using the LP Model Frequency Response

Spectral subtraction can be applied either on the short-time spectral amplitude of noisy speech obtained from DFT or on the magnitude of the frequency response of a linear prediction model of noisy speech, as illustrated in Figure 16.6.

For LP-based spectral subtraction (LPSS), the filter response $W_{LPSS}(k)$ is obtained from equations similar to Equations (16.14) and (16.15) with the main difference that, instead of the DFT-based magnitude spectrum, the LP-based magnitude spectrum of noisy speech and the average LP power spectrum of noise are used. LP spectral subtraction involves the following steps:

- (1) Obtain the coefficient vectors of the LP models of noisy speech and noise, and hence the magnitude frequency responses of the LP models of noisy speech, $Y_{LP}(k)$, and noise, $\hat{N}_{LP}(k)$.
- (2) Using the magnitude frequency responses of the LP models of speech and noise, obtain the frequency response of the spectral subtraction filter, $W_{LPSS}(k)$ from Equation (16.14).
- (3) Restore the speech signal through application of the smoothed LP spectral subtraction filter to the noisy speech.

16.2.4 BAYESIAN MMSE SPEECH ENHANCEMENT

The probabilistic minimum mean squared error estimation of the short-time spectral amplitude of speech is a Bayesian estimation method with a mean squared error cost function. The Bayesian estimation of spectral amplitude was obtained in Chapter 11 as

$$\hat{X}_k = \frac{\int_{-\infty}^{\infty} \int_0^{2\pi} X_k p[Y(k)|X_k, \theta_{X_k}] p(X_k, \theta_{X_k}) d\theta_{X_k} dX_k}{\int_{-\infty}^{\infty} \int_0^{2\pi} p[Y(k)|X_k, \theta_{X_k}] d\theta_{X_k} dX_k} \tag{16.16}$$

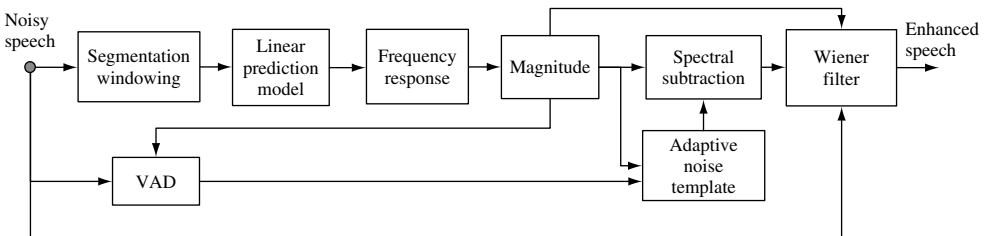


Figure 16.6 Block diagram illustration of LP-based spectral subtraction system for de-noising speech.

where $p[X(k)|Y(k)]$ is the posterior probability of clean speech, $X(k)$, given noisy observation $Y(k)$. The MMSE Equation (16.16) requires the likelihood of the noisy speech, $p[Y(k)|X_k, \theta_{X_k}]$, and the prior probability density function of the clean speech, $p(X_k, \theta_{X_k})$.

As described in Chapter 11, Ephraim and Malah derived an MMSE spectral amplitude estimation algorithm using a Rayleigh distribution for the magnitude spectrum of clean speech, a uniform distribution for the phase of the clean speech and a complex Gaussian distribution for noisy speech. The resulting estimator is of the form:

$$\hat{X}_k = W_{\text{MMSE}}(k)Y_k \tag{16.17}$$

where the gain factor, $W_{\text{MMSE}}(k)$, is given by

$$W_{\text{MMSE}}(k) = \Gamma(1.5) \frac{\sqrt{v_k}}{\gamma_k} \exp\left(-\frac{v_k}{2}\right) \left[(1 + v_k)I_0\left(-\frac{v_k}{2}\right) + v_k I_1\left(-\frac{v_k}{2}\right) \right] \tag{16.18}$$

where $\Gamma(\cdot)$ is the gamma function, $I_n(\cdot)$ is Bessel function of order n and v_k and γ_k are defined as

$$v_k = \frac{\xi(k)}{1 + \xi(k)} \gamma_k, \xi_k = \frac{\sigma_X^2(k)}{\sigma_N^2(k)}, \gamma_k = \frac{Y^2(k)}{\sigma_N^2(k)} \tag{16.19}$$

where $\sigma_X^2(k)$ and $\sigma_N^2(k)$ are the variance of speech and noise spectra, ξ_k is known as the prior signal-to-noise ratio and γ_k is known as the posterior signal-to-noise ratio.

Figure 16.7 shows a comparison of the performance of spectral subtraction based on LPSS and the MMSE method for de-noising of speech, observed in train noise.

16.2.5 KALMAN FILTER

The Kalman filter, described in Chapter 7, differs from the Wiener filter in several ways, the most important being that Kalman formulation permits explicit inclusion of time-varying equations for speech and noise processes. For de-noising speech, the Kalman filter can be implemented in the time or frequency domain. The conventional method is the time-domain Kalman filter.

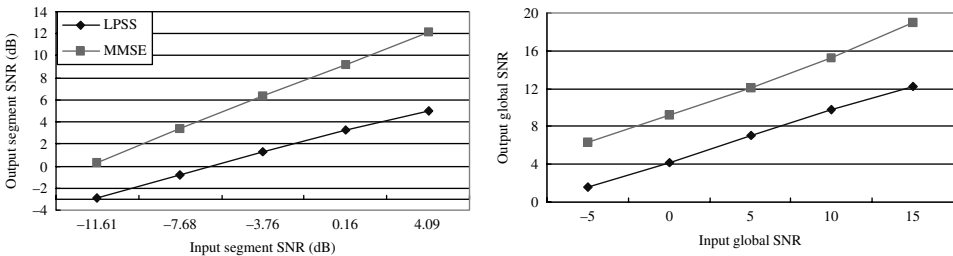


Figure 16.7 Performance comparison of LPSS and MMSE: output SNR vs input SNR.

16.2.5.1 Kalman State–Space Equations of Signal and Noise Models

The speech signal, $x(m)$, and the noise, $n(m)$, are modelled by autoregressive processes as

$$x(m) = \sum_{k=1}^P a_k x(m-k) + e(m) \quad (16.20)$$

$$n(m) = \sum_{k=1}^q b_k n(m-k) + v(m) \quad (16.21)$$

where a_k and b_k are the coefficients of autoregressive model of speech and noise, respectively. Equations (16.20) and (16.21) can be expressed in a state–space form for Kalman filtering as

$$\mathbf{x}(m) = \mathbf{A}_x \mathbf{x}(m-1) + \mathbf{g}_x e(m) \quad (16.22)$$

$$x(m) = \mathbf{h}_x \mathbf{x}(m) \quad (16.23)$$

$$\mathbf{n}(m) = \mathbf{A}_n \mathbf{n}(m-1) + \mathbf{g}_v v(m) \quad (16.24)$$

$$n(m) = \mathbf{h}_n \mathbf{n}(m) \quad (16.25)$$

where the AR model coefficient matrices are defined as

$$\mathbf{A}_x = \begin{bmatrix} a_1 & a_2 & \cdots & a_{p-1} & a_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{A}_n = \begin{bmatrix} b_1 & b_2 & \cdots & b_{q-1} & b_q \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (16.26)$$

The signal and noise vectors are $\mathbf{x}(m) = [x(m), x(m-1), \dots, x(m-p)]^T$, $\mathbf{n}(m) = [n(m), n(m-1), \dots, n(m-q)]^T$, $\mathbf{h}_x = \mathbf{g}_x = [1, 0, \dots, 0]^T$ and $\mathbf{h}_v = \mathbf{g}_v = [1, 0, \dots, 0]^T$. The signal and noise processes can be combined into a state–space equation as

$$\underbrace{\begin{bmatrix} \mathbf{x}(m) \\ \mathbf{n}(m) \end{bmatrix}}_{\mathbf{z}(m)} = \underbrace{\begin{bmatrix} \mathbf{A}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_n \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{x}(m-1) \\ \mathbf{n}(m-1) \end{bmatrix}}_{\mathbf{z}(m-1)} + \underbrace{\begin{bmatrix} \mathbf{g}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{g}_n \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} e(m) \\ v(m) \end{bmatrix}}_{\mathbf{w}(m)} \quad (16.27)$$

In compact notation, the state equation can be written as

$$\mathbf{z}(m) = \mathbf{A} \mathbf{z}(m-1) + \mathbf{G} \mathbf{w}(m) \quad (16.28)$$

16.2.5.2 Noisy Observation (Measurement) Equation

The noisy observation may be expressed as

$$y(m) = \mathbf{h}^T \mathbf{z}(m) = x(m) + n(m) \quad (16.29)$$

where

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_x^T \\ \mathbf{h}_n^T \end{bmatrix} \quad (16.30)$$

Note that, in this formulation of the Kalman filter, the signal and noise form parts of the state vector in Equation (16.27) and the noisy observation given by Equation (16.29) is a linear transformation of the state vector that simply produces addition of signal and noise. The Kalman filter equations given in Chapter 7 and adapted here are as follows:

- *the state vector prediction equation* (the state vector is composed of speech and noise)

$$\hat{\mathbf{z}}(m|m-1) = \mathbf{A}(m-1)\hat{\mathbf{z}}(m-1) \quad (16.31)$$

- *the covariance matrix of prediction error*

$$\mathbf{P}(m|m-1) = \mathbf{A}(m-1)\mathbf{P}(m-1)\mathbf{A}^T(m-1) + \mathbf{G}\mathbf{V}(m-1)\mathbf{G} \quad (16.32)$$

- *the Kalman gain vector*

$$\mathbf{K}(m) = \mathbf{P}(m|m-1)\mathbf{h}[\mathbf{h}^T\mathbf{P}(m|m-1)\mathbf{h}]^{-1} \quad (16.33)$$

- *the state update equation*

$$\hat{\mathbf{z}}(m) = \hat{\mathbf{z}}(m|m-1) + \mathbf{K}(m)[y(m) - \mathbf{h}^T\hat{\mathbf{z}}(m|m-1)] \quad (16.34)$$

Note that the innovation signal $y(m) - \mathbf{h}^T\hat{\mathbf{z}}(m|m-1)$ is a mixture of the unpredictable parts of the signal and the noise. The covariance matrix of estimation error is

$$\mathbf{P}(m) = (\mathbf{I} - \mathbf{K}\mathbf{h}^T)\mathbf{P}(m|m-1) \quad (16.35)$$

The application of Kalman filter requires estimates of the AR models of speech and noise, $\mathbf{a} = [a_1, a_2, \dots, a_p]$ and $\mathbf{b} = [b_1, b_2, \dots, b_q]$ (Figure 16.8). These are obtained from an

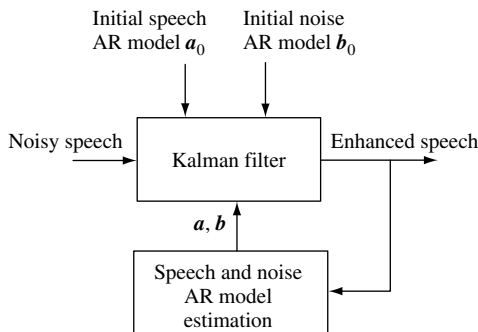


Figure 16.8 Illustration of the Kalman filter method of enhancement of noisy speech.

application of the estimate–maximize algorithm, which effectively yields the following normal equation (the Yule–Walker equation):

$$\hat{\mathbf{a}} = \mathbf{R}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}^{-1} \mathbf{r}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \quad (16.36)$$

$$\hat{\mathbf{b}} = \mathbf{R}_{\hat{\mathbf{n}}\hat{\mathbf{n}}}^{-1} \mathbf{r}_{\hat{\mathbf{n}}\hat{\mathbf{n}}} \quad (16.37)$$

where the autocorrelation matrices of speech and noise are obtained from Kalman estimates as

$$\mathbf{R}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = E \{ [\hat{\mathbf{x}}(m), \dots, \hat{\mathbf{x}}(m-p)]^T [\hat{\mathbf{x}}(m), \dots, \hat{\mathbf{x}}(m-p)] \} \quad (16.38)$$

$$\mathbf{R}_{\hat{\mathbf{n}}\hat{\mathbf{n}}} = E \{ [\hat{\mathbf{n}}(m), \dots, \hat{\mathbf{n}}(m-q)]^T [\hat{\mathbf{n}}(m), \dots, \hat{\mathbf{n}}(m-q)] \} \quad (16.39)$$

Figure 16.8 is an outline illustration of Kalman filter method of enhancement of noisy speech.

16.2.6 SPEECH ENHANCEMENT VIA LP MODEL RECONSTRUCTION

Speech can be enhanced through a process of decomposition, de-noising and then reconstruction of the source-filter parameters of a linear prediction model of speech. An LP model of speech, described in detail in Chapter 8, may be expressed as

$$X(z) = E(z)V(z) \quad (16.40)$$

where $E(z)$ is the z -transform of the excitation signal and $V(z)$ is the z -transfer function of a combined model of the vocal tract, glottal pulse and lip radiation; $V(z)$ can be expressed by a cascade combination of a set of second-order resonators and a first-order model as

$$V(z, m) = G(m) \frac{1}{1 + r_0(m)z^{-1}} \prod_{k=1}^N \frac{1}{1 - 2r_k(m)\cos(\phi_k(m))z^{-1} + r_k^2(m)z^{-2}} \quad (16.41)$$

where $\phi_k(m)$ and $r_k(m)$ are the time-varying radii and the angular frequencies of the poles of the LP model, respectively, and $G(m)$ is the overall gain of the LP model. In Equation (16.41) the second-order sections model the resonance of speech spectrum and the first-order section models the slope of the spectrum of speech.

Formants are the resonances of speech. The poles of the LP model are the formant candidates, the raw data from which formants and their models are estimated. The spectral resonance at a formant is characterized by a parameter vector comprising the frequency F_k , bandwidth B_k and magnitude of the resonance M_k and their temporal slope of variation (velocity), as

$$\mathbf{F}_k = [F_k, B_k, M_k, \Delta F_k, \Delta B_k, \Delta M_k] \quad k = 1, \dots, M \quad (16.42)$$

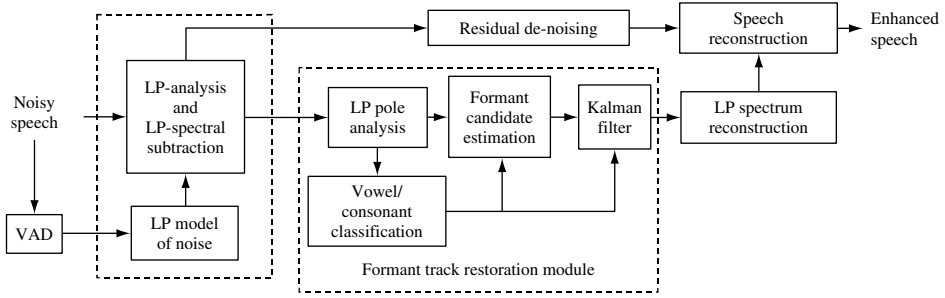


Figure 16.9 Overview of the speech-enhancement system.

where Δ denotes the slope of the trajectory of a feature vector over time, e.g. $\Delta F_k(t)$ representing the velocity of the k th formant at frame t is obtained as

$$\Delta F_k(t) = \frac{\sum_{m=1}^N m [F_k(t+m) - F_k(t-m)]}{\sum_{m=1}^N 2m^2} \quad k = 1, \dots, M \quad (16.43)$$

Equations (16.41)–(16.43) lead to the concept of formant-tracking LP models for speech restoration. Format-synthesisers are well established in speech technology, many being developments of the Klatt synthesiser.

16.2.6.1 Formant-tracking Speech Restoration System

The formant-tracking speech restoration system, illustrated in Figure 16.9, consists of the following sections:

- (1) an LP model analysis module for estimation of the LP model parameters and the excitation parameters of noisy speech;
- (2) a noise detection–estimation module for estimation of the time-varying parameters of noise model;
- (3) a speech pre-cleaning module based either on the spectral subtraction or on the MMSE spectral amplitude estimation;
- (4) a formant tracking method for tracking the formant parameters of pre-cleaned speech across successive speech frames;
- (5) an excitation model for estimating the excitation parameters and providing an estimate of the cleaned excitation;
- (6) speech enhancement via synthesis using an estimate of the LP model of speech combined with excitation.

The input to the LP pole analysis is the noisy speech spectrum with the mean of the LP spectrum of noise removed through spectral subtraction. An LP model is fitted to the speech

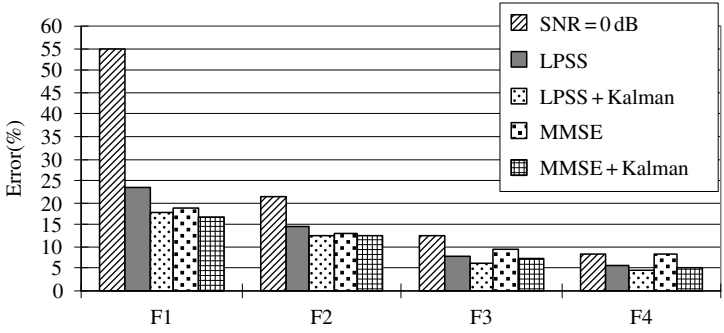


Figure 16.10 Average percentage error of formant tracks of speech in train noise and cleaned speech using LPSS, MMSE and Kalman filters; the results were averaged over five male speakers.

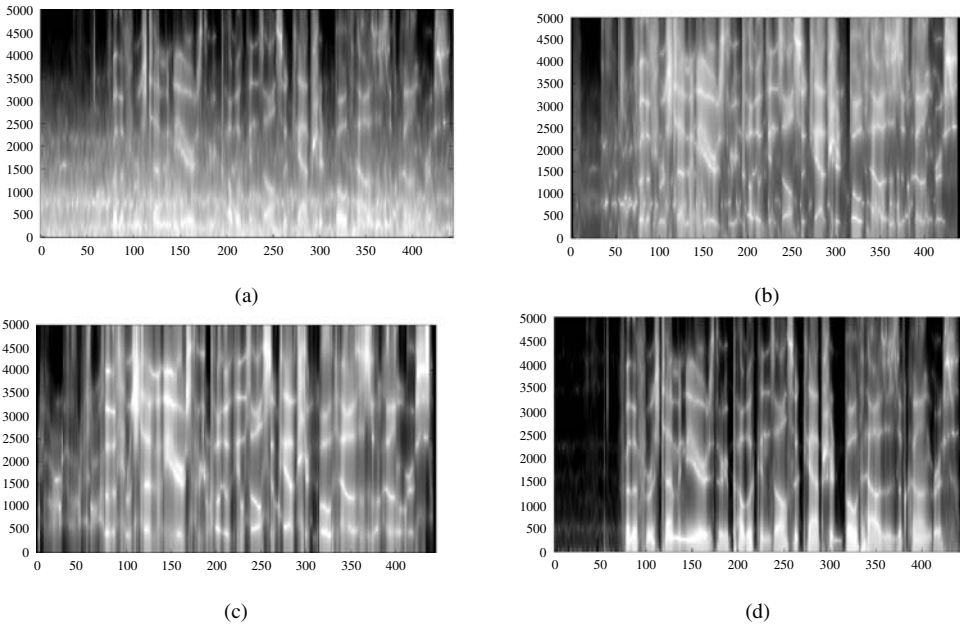


Figure 16.11 Comparison of LP spectrograms from: (a) speech in train noise; (b) pre-cleaned speech; (c) formant synthesized speech; and (d) clean speech.

spectrum and the poles of the model are obtained through a polynomial rooting function. Figure 16.10 shows the reduction in formant tracking error from application of different noise reduction methods. Figure 16.11 shows a comparative illustration of the LP spectrograms of clean speech, noisy speech and speech restored with the spectral subtraction method and the formant synthesis method.

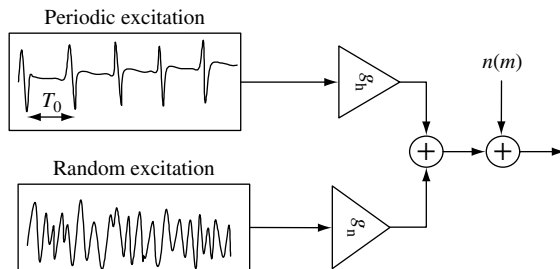


Figure 16.12 A harmonic plus noise model of speech excitation observed in background noise.

16.2.6.2 De-noising of Speech Excitation Signal

The noisy speech excitation can be modeled, Figure 16.12, as a combination of the harmonic and the fricative noise content of the excitation and the residual background noise remaining after the inverse LP filtering as

$$e(m) = g_h \sum_{k=1}^N [a_k \sin(2\pi k F_0(m)) + b_k \cos(2\pi k F_0(m))] + g_n v(m) + n_r(m) \quad (16.44)$$

where $F_0(m)$ is the time-varying fundamental frequency of speech excitation, a_k and b_k are the amplitude of each excitation harmonic, $v(m)$ is the noise-like excitation, g_h and g_n are the mixture weights and $n_r(m)$ is the background noise after inverse LP filtering.

16.3 MULTIPLE-INPUT SPEECH-ENHANCEMENT METHODS

In multiple-input noise-reduction systems several noisy input signals, picked up by a array of microphones, are filtered and combined to reduce noise, echo, interfering speech, room reverberations and distortions.

Multiple-input speech-enhancement systems include adaptive beam forming, adaptive noise cancellation, MIMO teleconferencing systems, stereophonic echo cancellation and in-car cabin communication systems.

In a typical multiple-input system, Figure 16.13, there are several microphones. The output of each microphone is a mixture of the speech signal, feedback from loudspeakers, speech reflections from walls and noise.

Assuming that there are M microphones and N sets of signal and noise sources, then there are $N \times M$ different acoustic channels between the sources of signals and the microphones. We can write a system of linear equations to describe the relationship between the signals emitted from different sources, $x_i(m)$, and the signals picked up by the microphones $y_j(m)$ as

$$y_j(m) = \sum_{i=1}^N \sum_{k=0}^P h_{ij}(k) x_i(m - k) \quad j = 1, \dots, M \quad (16.45)$$

where $h_{ij}(k)$ denotes the response of the channel from the source i to the microphone j modelled by a finite impulse response filter. Note that, for simplicity, each source of signal,

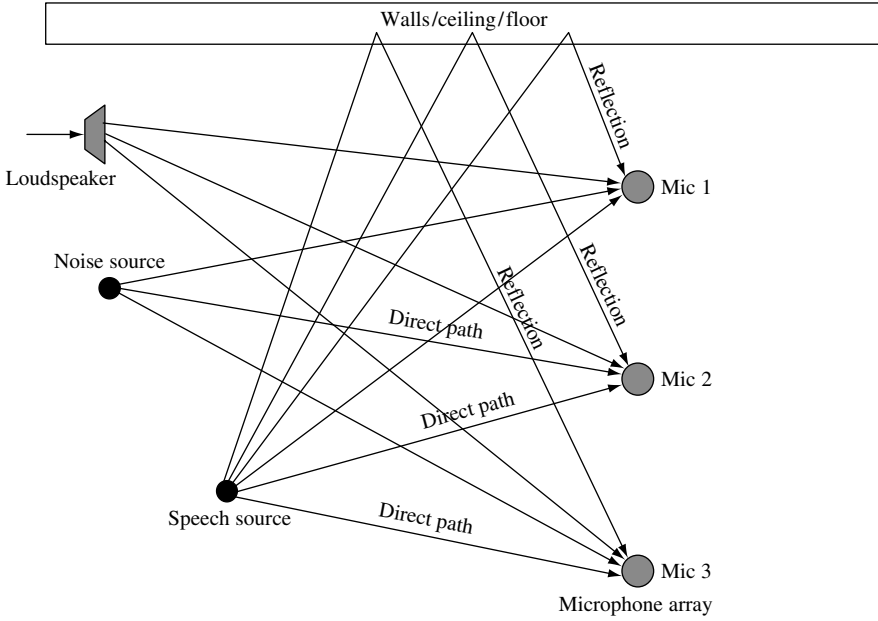


Figure 16.13 Illustration of different sounds and noise arriving at microphones, via direct line of sight paths and via reflections from walls/ceiling/floor.

noise or interference is denoted with same letter x and different index, as $x_i(m)$, m being the discrete-time index.

In the simplest MIMO model, the response of an acoustic channel from a sound source i to a microphone j via a direct or reflected path can be represented by two parameters: an attenuation factor, $\alpha_{ij}(m)$, and a propagation time delay, $\tau_{ij}(m)$, as

$$h_{ij}(m) = \alpha_{ij}(m)\delta [m - \tau_{ij}(m)] \tag{16.46}$$

Note that each source of sound may reach a microphone via a direct path and via a number of indirect paths after reflections, in which case the response from a source i to a microphone j needs to be expressed as

$$h_{ij}(m) = \sum_{k=1}^L \alpha_{ijk}(m)\delta [m - \tau_{ijk}(m)] \tag{16.47}$$

where $\alpha_{ijk}(m)$ and $\tau_{ijk}(m)$ are the attenuation factor and the propagation time delay along the k th path from source i to microphone j .

MIMO speech enhancement systems are subjects of much continuing research. In general, the main issues in MIMO noisy speech processing systems are as follows:

- (1) identification of the channel responses or the room transfer functions, $\{h_{ij}(m)\}$, from the speech and/or noise sources to microphones;

- (2) the problem of nonuniqueness of the solutions for room channel responses, when the noise sources are correlated; this is explained in the stereo acoustic echo cancellation in Chapter 14;
- (3) the speed of convergence of adaptation of the filters coefficients at the output of each microphone to achieve cancellation of noise and acoustic feedback echo;
- (4) estimation of the time-varying speech and noise characteristics for noise suppression.

There are many different ways in which the outputs of the MIMO systems can be processed. One of the most promising applications of the MIMO system is the beam-forming microphone array, described next.

16.3.1 BEAM-FORMING WITH MICROPHONE ARRAYS

Beam-forming is a noise reduction, or noise screening, method in which an array of microphones and adaptive filters is employed to selectively seek a ‘beam’ of signal from a particular direction (usually the direction towards a desired source and where the signal from that source is strongest) and suppress noise, reverberations and acoustic feedback coming from other directions.

Beam-forming uses temporal and spatial filtering to create directional spatial selectivity. In this way it is possible to adaptively adjust the filters to selectively pick up a sound wave from the direction of the source where the sound energy is strongest and screen out noise, feedback and reflections of sounds from other directions. Beam-forming has applications in hands-free communications, such as in-car communications, personal computer voice communication, teleconferencing and robust speech recognition. Beam-forming can also be combined with acoustic feedback cancellation.

16.3.1.1 Principles of Microphone Array Beam-forming

Beam-forming relies on use of delays (or equivalently phase) for constructive and destructive combination of the incoming signals arriving at the microphones from different directions. Figure 16.14(a) shows the sound field of an omnidirectional microphone that picks up sound equally from all directions. The combined output of several microphones placed in an array exhibits directional selectivity, as shown in Figure 16.14(b).

The principle of beam-forming can be illustrated by considering a linear microphone array in the far field of sound source, as illustrated in Figure 16.15. The assumption that the

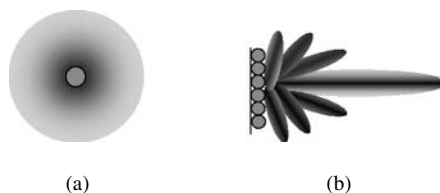


Figure 16.14 Illustration of the reception fields (aperture) of (a) an omnidirectional microphone and (b) an array of microphones.

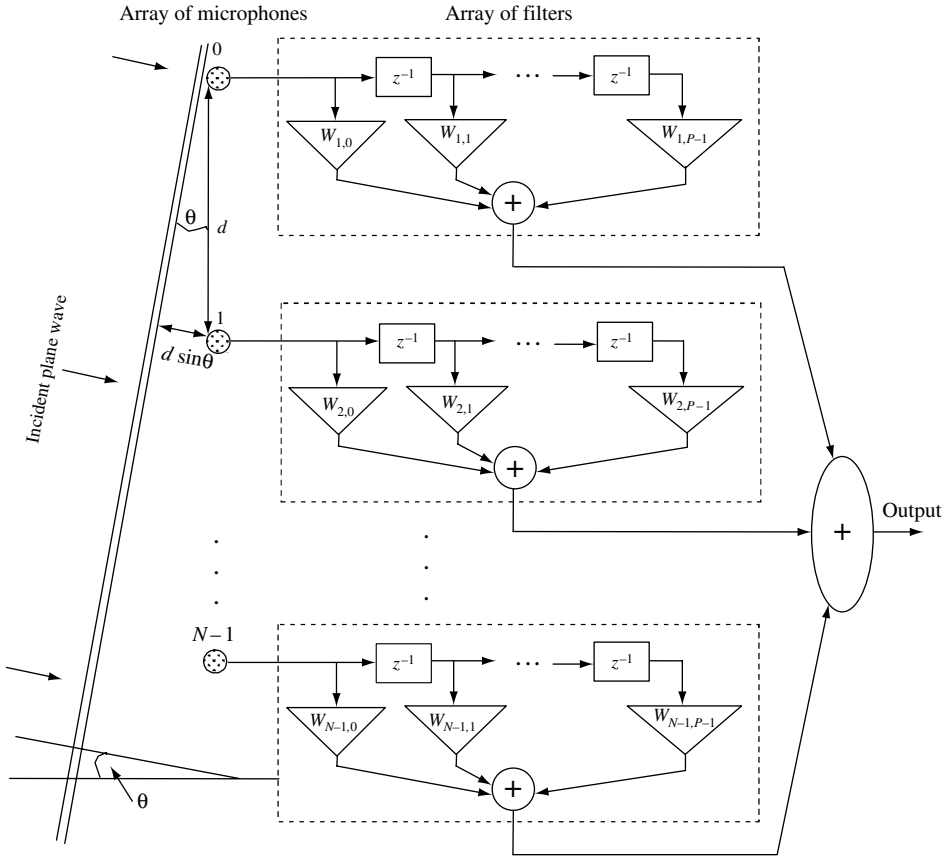


Figure 16.15 Illustration of beam-forming. The array of filters can be adjusted to change the ‘looking’ direction of the beam.

microphone is in the far field of the sound source has the useful implication that the sound waves reaching the microphone array are planar (as opposed to spherical).

Sound waves arriving at the microphone array, at an incident angle of θ , reach successive microphones at different times due to the different distances that the waves have to travel to reach different microphones. The additional distance that a wave has travelled to reach an adjacent microphone is $d \sin(\theta)$. Note that, if θ is zero, then the wave is perpendicular to the microphone array and in that case all microphones receive each plane wave at the same time.

Sound propagates at a speed of $v = 342 \text{ m/s}$ at a room temperature of 25°C . The additional time taken for sound to travel a distance of $d \sin(\theta)$ to reach an adjacent microphone is given by

$$\tau = \frac{d \sin(\theta)}{v} \tag{16.48}$$

For a sine wave it is easy to see that, if the distance between two microphones and the angle of incidence of the wave, θ , result in a time delay of the arrival of sound between the two adjacent microphones of say half a wavelength, then the sum of the outputs of the microphones will cancel out. Note that this illustration is oversimplified and in practice a combination of temporal and spatial filtering provides spatial selectivity. In general, signals arriving at a microphone array at the same time (in phase) add up, whereas signals arriving at microphone array at different times cancel out partially or completely.

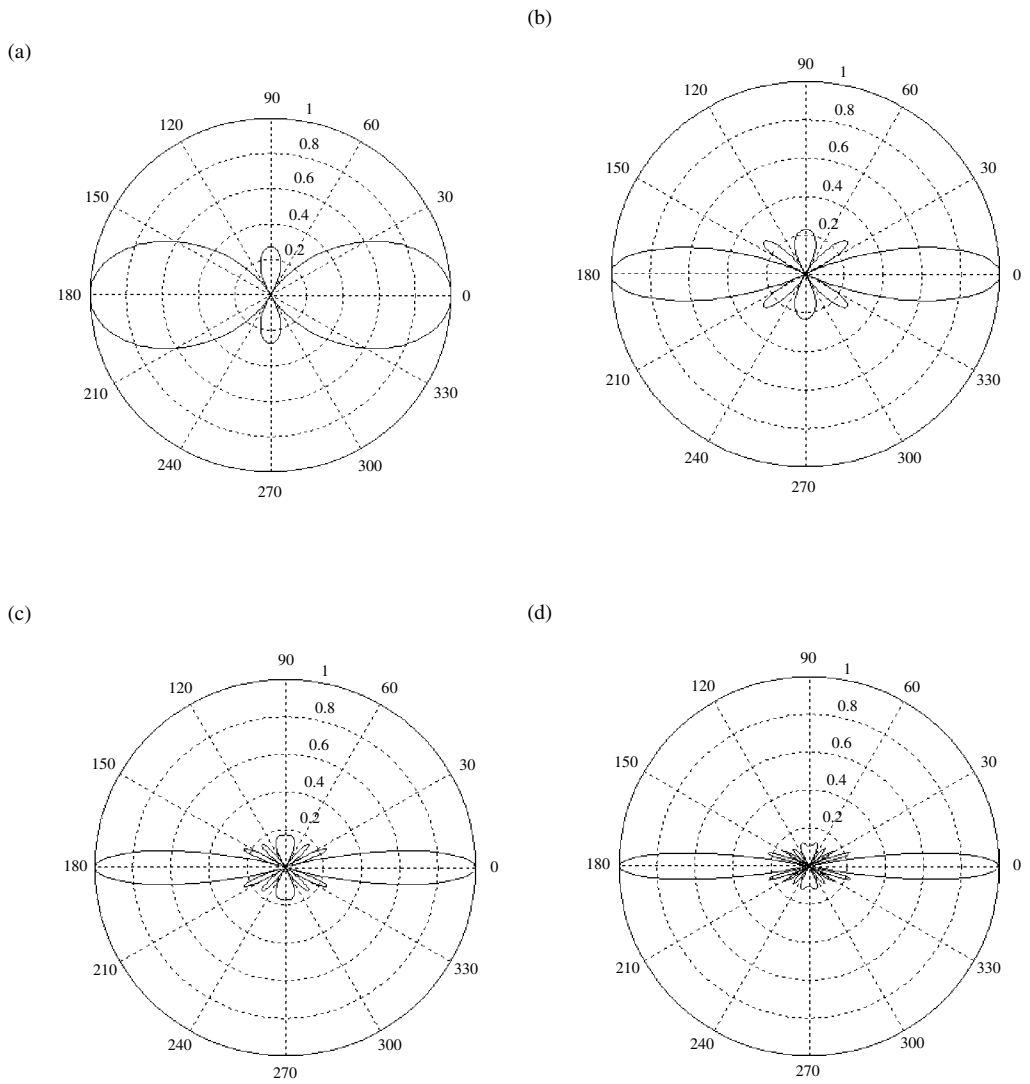


Figure 16.16 Illustration of the beam of a microphone array in a response to a frequency of 2 kHz with the microphones spaced at $d = 10$ cm. (a) Two, (b) four, (c) six and (d) eight microphones.

The combined output of N microphones can be expressed as

$$y(m) = \sum_{i=0}^{N-1} x(m - \tau_i) \quad (16.49)$$

where for simplicity we assume the channel attenuation factors $\alpha_i = 1$. The response of the microphone array to a frequency f and direction of arrival of signal θ can be expressed as

$$\begin{aligned} H(f, \theta) &= \sum_{m=0}^{N-1} e^{-j2\pi f m \tau} \\ &= \sum_{m=0}^{N-1} e^{-j2\pi f m [d \sin(\theta)/v]} \end{aligned} \quad (16.50)$$

Equation (16.50) is used to plot the response of a microphone array for a selected frequency and varying direction of arrival (Figure 16.16). The angle of arrival can be changed or adapted using the adaptive filters shown in Figure 16.15.

Note that the bandwidth of the beam decreases with the increasing number of microphones, but it is also a function of the distance between microphones and the frequency of the sound.

16.4 SPEECH DISTORTION MEASUREMENTS

The most commonly used measure for quality of speech is signal-to-noise ratio. An average SNR measure is defined as

$$\text{SNR} = 10 \log_{10} \left(\frac{P_{\text{Signal}}}{P_{\text{Noise}}} \right) \text{ dB} \quad (16.51)$$

where P_{Signal} and P_{Noise} are the power of signal and noise, respectively.

The segmental SNR is defined as

$$\text{SNR}_{\text{seg}} = \frac{1}{K} \sum_{k=0}^{K-1} 10^* \log_{10} \left\{ \frac{\sum_{m=0}^{N-1} x_k^2(m)}{\sum_{m=0}^{N-1} [x_k(m) - \hat{x}_k(m)]^2} \right\} \text{ dB} \quad (16.52)$$

where $x_k(m)$ and $\hat{x}_k(m)$ are the clean signal and restored signal at frame m , N is the total number of frames and K is the number of samples in each frame. The segmental SNR of speech signals can fluctuate widely, as illustrated in Figure 16.15, which shows the variation of segmental SNR at average SNRs of 0, 5 and 10 dB.

The signal-to-noise ratio is not the best measure of speech quality as it does not take into account the structure of the speech or the psychoacoustics of hearing.

The Itakura–Saito distance (ISD) measure is defined as

$$\text{ISD}_{12} = \frac{1}{N} \sum_{j=1}^N \frac{[a_1(j) - a_2(j)] R_1(j) [a_1(j) - a_2(j)]^T}{a_1(j) R_1(j) a_1(j)^T} \quad (16.53)$$

where $a_1(j)$ and $a_2(j)$ are the linear prediction model coefficient vector calculated from clean and transformed speech at frame j and $R_1(j)$ is an autocorrelation matrix derived from

the clean speech. Owing to the asymmetry of the ISD measure (i.e. $ISD_{21} \neq ISD_{12}$), the following segmental ISD measure is used:

$$ISD_{\text{sym}} = (ISD_{12} + ISD_{21})/2 \quad (16.54)$$

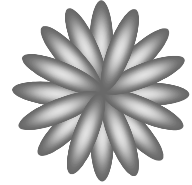
The ISD criterion is a more balanced measure of the distance between an original clean speech signal and a distorted speech signal as speech frames with relatively large SNRs do not dominate the overall distance measure to the same extent as in the more conventional SNR measures of Equations (16.51)–(16.52).

BIBLIOGRAPHY

- Bakamidis S., Dendrinis M. and Carayannis G. (1991) Speech enhancement from noise: a regenerative approach. *Speech Commun.*, **10**: 44–57.
- Berouti M., Schwartz R. and Makhoul J. (1979) Enhancement of speech corrupted by acoustic noise. *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 208–211.
- Boll S.F. (1979) Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Process.*, **ASSP-27**: 113–120
- Cohen I. (2004) Speech enhancement using a noncausal *a priori* SNR estimator. *IEEE Signal Process. Lett.*, **11**(9): 725–728.
- Deller J.R., Proakis J.G. and Hansen, J.H. (1993) *Discrete-time Processing of Speech Signals*. Macmillan, New York.
- De Moor B. (1993) The singular value decomposition and long and short spaces of noisy matrices. *IEEE Trans. Signal Process.*, **41**(9): 2826–2838.
- Ephraim Y. (1992) Statistical-model based speech enhancement systems. *Proc. IEEE*, **80**(10): 1526–1554.
- Ephraim Y. and Malah D. (1985) Speech enhancement using a minimum mean square error log-spectral amplitude estimator. *IEEE Trans. Acoust., Speech, Signal Process.*, **ASSP-33**: 443–445.
- Ephraim Y. and Van Trees H.L. (1995) A signal subspace approach for speech enhancement. *IEEE Trans. Speech Audio Process.*, **3**(4): 251–266.
- Gannot S., Burshtein D. and Weinstein E. (1998) Iterative and sequential Kalman filter-based speech enhancement. *IEEE Trans. Speech Audio Process.*, **6**(4): 373–385.
- Hansen J.H.L. and Clements M.A. (1987) Iterative speech enhancement with spectral constraints. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing*, pp. 189–192.
- Kalman R. (1960) A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Engng*, **82**: 34–35.
- Lim J.S. and Oppenheim A.V. (1978) All-pole modeling of degrade speech. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-26**(3): 197–210
- Lim J.S. and Oppenheim A.V. (1979) Enhancement and bandwidth compression of noisy speech. *Proc. IEEE*, **67**: 1586–1604.
- Mack G.A. and Jain V.K. (1985) A compensated-Kalman speech parameter estimator. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing*, pp. 1129–1132.
- Martin R. (2002) Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing*, Orlando, FL.
- Martin R. and Breithaupt C. (2003) Speech enhancement in the DFT domain using Laplacian speech priors. *Proceedings of International Workshop Acoustic Echo and Noise Control (IWAENC)*, 2003, pp. 87–90.
- McAulay R.J. and Malpass M.L. (1980) Speech enhancement using a soft-decision noise suppression filter. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-28**: 137–145.
- McCandles S.S. (1974) An algorithm for automatic formant extraction using linear prediction spectra. *IEEE Trans. Acoust. Speech Signal Process.*, **22**: 135–141.
- Paliwal K.K. and Alsteris, L.D. (2003) Usefulness of phase spectrum in human speech perception. *Proceedings of European Conference on Speech Communication and Technology (Eurospeech-03)*, Geneva, pp. 2117–2120.
- Paliwal K.K. and Basu A. (1987) A speech enhancement method based on Kalman filtering. *Proceedings of IEEE, International Conference on Acoustics, Speech and Signal Processing*, pp. 177–180.

- Rabiner L.R. (1989) A tutorial on hidden Markov models and selected application in speech recognition. *IEEE Proc.*, **77**(2): 257–286
- Rabiner L.R. and Juang B.H. (1993) *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ.
- Rabiner L.R. and Schafer R.W. (1978) *Digital Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, NJ.
- Rentzos D., Vaseghi S., Yan Q., Ho C. and Turajlic E. (2003) Probability models of formant parameters for voice conversion. *Proceedings of Eurospeech*, pp. 2405–2408.
- Rigoll G. (1986) A new algorithm for estimation of formant trajectories directly from the speech signal based on an extended Kalman filter. *Proceedings of ICASSP*, pp. 1229–1232.
- Sameti H., Sheikhzadeh H., Deng L. and Brennan R.L. (1998) HMM-based strategies for enhancement of speech signals embeded in non-stationary noise. *IEEE Trans. Speech Audio Process.*, **6**(5): 445–455.
- Schroeder, M.R. (1999) *Computer Speech: Recognition, Compression, Synthesis*. Springer, Berlin.
- Shafer, R.W. and Rabiner L.R. (1970) System for automatic formant analysis of voiced speech. *J. Acoust. Soc. Am.*, **47**(2): 634–650.
- Wang, D.L. and Lim, J.S. (1982) The unimportance of phase in speech enhancements. *IEEE Trans. Acoust. Speech Signal Process.*, **30**: 679–681.
- Weber K., Bengio S. and Bourlard H. (2001) HMM2-extraction of formant structures and their use for robust ASR. *Proceedings of Eurospeech*, pp. 607–610.

17



Noise in Wireless Communications

17.1 Introduction to Cellular Communications	434
17.2 Noise, Capacity and Spectral Efficiency	436
17.3 Communications Signal Processing in Mobile Systems	438
17.4 Noise and Distortion in Mobile Communications Systems	439
17.5 Smart Antennas	444
17.6 Summary	447
Bibliography	448

Noise, fading and limited radio bandwidth are the main factors that limit the capacity and speed of communication on wireless channels. Much of the research and development in the design of new generations of mobile communications systems aim to increase the spectral efficiency, defined as the number of bits per second per Hertz of the bandwidth of a communications channel.

For improved efficiency, modern mobile communications systems rely on digital signal processing methods at almost every stage from source coding (e.g. code-excited linear prediction speech coders, MP3 music coders) to the allocation of time and bandwidth resources (e.g. code division multiple access systems). Furthermore, the capacity and performance of mobile communications systems is noise-limited and the use of signal processing algorithms in noise reduction, channel equalisation, echo cancellation, error control coding and more recently in space–time signal processing using smart beam-forming antennas, is playing an increasingly pivotal role in the development of efficient and intelligent communications systems. In this chapter we consider how communications signal processing methods are used to improve the speed and capacity of communications systems.

17.1 INTRODUCTION TO CELLULAR COMMUNICATIONS

The age of radio communications began in the 1860s with James Clark Maxwell's development of the electromagnetic wave theory. Maxwell predicted the existence of electromagnetic radio waves with various frequencies propagating with the speed of light and concluded that light itself was an electromagnetic wave.

In 1884 Heinrich Rudolf Hertz reformulated Maxwell's equations. Then between 1885 and 1888, in series of experiments, Hertz demonstrated that a rapidly oscillating electric current could be launched into space as an electromagnetic wave and detected. For generating and transmitting radio waves Hertz used a high-voltage induction coil, a capacitor, a rod with a gap in the middle and a spark sphere attached to the rod on each side of the gap. This device created oscillatory sparks as the spheres charged and discharged with opposite-polarity electric charges.

To detect the electromagnetic radiation, Hertz used a copper wire bent into a loop, with a small brass sphere connected to one end while the other end of the wire loop that pointed to the sphere had a screw mechanism so that the point could be adjusted and moved closer to the sphere. The presence of an oscillating charge in the receiver caused sparks across the very small gap between the end points of the wire loop. Hertz also demonstrated that radio waves had all the well-known properties of light waves – reflection, refraction, interference and polarization. It is said that, in response to questions from the students who witnessed his classroom experiments on generation and reception of electromagnetic waves, Hertz said that he saw no practical use for electromagnetic waves.

In 1895 Guglielmo Marconi, the inventor of the radiotelegraph, was inspired by Hertz's experiments and demonstrated the feasibility of radio communication by transmitting and receiving the first radio signal in Italy. By 1899 Marconi had telegraphed the first wireless signal across the English Channel and in 1901 he sent the first transatlantic radiotelegraph message from England to Newfoundland. Note, however, that in 1943 the US Supreme Court overturned Marconi's patent in favour of Nicola Tesla, who is now credited with invention of the radio. Another scientist who is also credited with the invention of radio is Jagadis Chandra Bose, who demonstrated his radio in 1896.

From the beginning of the radio age, the quality of radio transmission and robustness to noise and fading have been the major challenges in the design of wireless communications systems. High-quality radio transmission was made possible by Edwin Howard Armstrong's invention of the wideband frequency modulation (FM) in 1933. The FM method can trade off more bandwidth for more robustness to noise and provide transparent broadcast quality. The FM modulation remained the main method for high-quality civilian radio transmission until the recent deployment of digital radio communications.

The era of mobile cellular communication began in 1970s when AT&T proposed the first high-capacity analogue telephone system, called the advanced mobile phone service (AMPS). This system was later extended to a cellular mobile system. Over the past three decades cellular mobile phone technology has developed rapidly through the advent of digital mobile phone standards such as the European Global Mobile System (GSM) standard and the American IS-95 and IS-96 standards. In the late 1990s, the increasing demand for mobile phone and multimedia services led to the development of 3G and UMTS standards.

The main limited resources in modern mobile communications systems are bandwidth and on-board battery power. The available radio bandwidth is limited by the need to share the

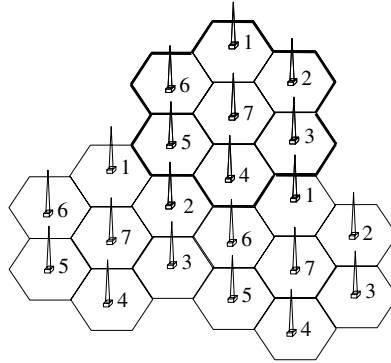


Figure 17.1 A cellular system increases capacity by dividing a geographical area into clusters of cells and reusing the radio spectrum in nonadjacent cells. Each set of radio channel frequencies used in a cell is shown by a different number.

finite radio frequency channels among many different mobile users and for many different applications and purposes. The available power is limited by the capacity, size and weight of the on-board batteries. In a mobile communications system the radio frequency bandwidth needs to be used efficiently in order to maximize the capacity.

Figure 17.1 depicts the geometric topology of a cellular mobile radio system. In cellular systems a city or a town is divided into a number of geographical cells. The cells are thought of as having a hexagonal shape. A key aspect of the cellular mobile communications technology is the very large increase in spectral efficiency achieved through the arrangements of cells in clusters and the *reuse* of the radio frequency channels in nonadjacent cells; this is possible because the cell phones and base stations operate on low-power transmitters whose electromagnetic wave energy fades before it reaches nonadjacent cells. In the cellular configuration of Figure 17.1, each cell within each cluster of seven cells uses a set of different frequencies (each set of radio channel frequencies used in a cell is shown by a different number). Different clusters reuse the same sets of frequencies, as shown by the number codes in each cluster of cells.

Each cell has a base station that accommodates the transmitter/receiver antennas, the switching networks and the routing equipment of the base station. Figure 17.2 shows the basic network architecture of a GSM cellular mobile phone system. It consists of the following subsystems.

- BTS – the base transceiver station is composed of transmitter/receiver antennas;
- BSC – the base station controller is responsible for allocation of the radio channels, terrestrial channel management, mapping of the radio channels onto wired channels and execution of the hand-over function as mobile users move across different cells;
- MSC – the mobile switching centre is responsible for the switching/routing of calls;
- HLR – the home location register is a database that holds general information about subscribers;
- VLR – the visitor location register is a temporary database that holds visiting subscribers' general information during their visit; it also holds location area identities of roaming users;

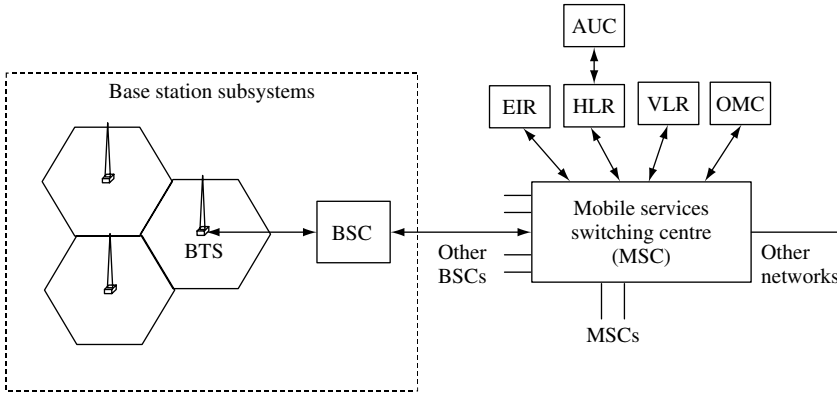


Figure 17.2 Illustration of a basic GSM mobile phone network.

- EIR – the equipment and identity register is a database that holds user equipment identities;
- OMC – the operation and maintenance centre is a database that holds relevant information about the overall operation and maintenance of the network.

17.2 NOISE, CAPACITY AND SPECTRAL EFFICIENCY

A principal challenge in the development of mobile communication systems stems from the limitations on data transmission due to finite bandwidth and the physical properties of the communications channels, which can include signal dispersion, fading, impulsive noise, co-channel/multiple-access interference as well as other phenomena such as the multipath effect.

The rate at which binary data bits (i.e. sequences of ones and zeros) can be transmitted on a communications channel is limited by (a) the available bandwidth and (b) the noise and interference.

Binary data are signalled using communication symbols. In its simplest form a communication symbol may be a pulse that takes two different amplitude levels for signalling a binary data bit. The maximum rate at which communication symbols can be transmitted is limited by the bandwidth. The symbol rate, r_s , that is the number of symbols per second of transmission, is about half the bandwidth.

However, each communication symbol may carry M bits provided that there are 2^M resolvable patterns of variation of the symbol such as the amplitude, phase or time delay (position) or frequency of the symbol. Assuming that each symbol carries M bits, the bit rate is M times the symbol rate, i.e. $r_b = Mr_s$. The maximum number of bits that can be signalled by each symbol is limited by noise and acceptable delay.

In practice the number of bits that each symbol can carry in an M -ary signalling scheme is limited by noise, interference, the multipath effect, channel distortions, echo and fading. The constraints imposed on the channel capacity due to noise and bandwidth limit the rate at which information can be transferred, even when multilevel encoding techniques are used. This is because noise obliterates the fine differences that distinguish the

various signal levels, limiting in practice the number of detection levels we can use in the decoder.

Therefore, from the above argument, the capacity of a communication link, that is the rate at which data can be transmitted, is proportional to the available bandwidth and the signal-to-noise ratio, the limit of which, in additive white Gaussian noise (AWGN), is expressed by the *Shannon–Hartley theorem* as

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (17.1)$$

where C is the channel capacity in bps, B is the channel bandwidth in Hz, and S/N is the signal-to-noise ratio expressed as a linear (as opposed to logarithmic) power ratio. The theorem, proved by Claude Shannon in 1948, describes the maximum capacity of a communication channel with error-correcting methods vs the levels of noise and interference. Equation (17.1) gives the theoretical limit that the best possible coding and modulation method may achieve. Figure 17.3 shows a plot of the bit rate per Hz vs signal-to-noise ratio. Note that, at an SNR of 0 dB, the maximum theoretical bit rate is 1 bit/Hz, at 15 dB it is 5 bits/Hz and this approaches a theoretical rate of 13 bits/Hz at 40 dB SNR.

The challenge in communications signal processing is to increase the capacity in bps/Hz of bandwidth through reducing noise, interference and multipath distortion. Signal processing methods play the central role in removing or compensating for noise and thereby improving capacity.

The signal processing methods that are used to improve the capacity of mobile communications systems include source coding, channel coding, channel equalisation, echo cancellation, multipath models and multiple access methods, including space–time signal processing via beam-forming antenna arrays. The use of multiple receiving and transmitting antennas combined with multiple access and noise-reduction methods opens myriad possibilities in the use of signal processing for enhanced mobile communications.

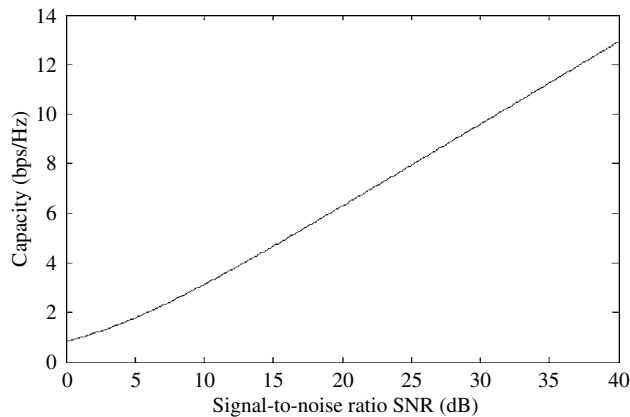


Figure 17.3 Illustration of the variation of the maximum capacity (bps/Hz) of a communication channel with the SNR (dB).

17.3 COMMUNICATIONS SIGNAL PROCESSING IN MOBILE SYSTEMS

The radio frequency bandwidth available to each operator of a cellular communication service is usually limited to a range of about several hundred kHz to several MHz centred between 1 and 2 GHz. The very large number of subscribers using mobile communications devices and the ever-increasing demand on the bandwidth is accommodated through the efficient use of communications resources that results in a large increase in the capacity per Hz also known as the spectral efficiency, defined as the data rate in bps/Hz unit of bandwidth:

$$\text{spectral efficiency} = \text{channel throughput/channel bandwidth}$$

Depending on the efficiency of the communications systems and the signal-to-noise ratio, the actual spectral efficiency may vary from 0.1 to 4 bps/Hz.

Modern mobile communications systems rely on advanced signal processing methods for fast, efficient, reliable and low-cost communications. The signal processing functions in a mobile communications system include the following modules:

- (1) Source coder/decoder – source coders compress signals at the transmitter by removing the correlation and redundancies from the signals; source decoders decompress and reconstruct the signals at the receiver. Source coding involves the use of signal transforms such as discrete Fourier transform and cosine transform, signal generation models such as linear prediction models and signal probability models such as entropy models, for the compression of the source data, which may include voice, image, video and text. Source coding methods, such as Huffman coders, MPEG music coders and CELP voice coders, can greatly reduce (often by a factor of 10–20) the required bit rate and bandwidth and hence increase the capacity and speed of transmission of audio, image, text and other data.
- (2) Channel coder/decoder – The purpose of channel coding is to reduce transmission errors due to noise, fading and loss of data packets. Channel coding involves the use of convolution and block coders for the addition of error-control bits to the source data in order to increase the distance between the sequences of transmitted data and hence improve the error detection and error correction capability of communication systems.
- (3) Multiple access signalling – this, as the name implies, provides simultaneous access to multiple users on the same shared bandwidth resource. Multiple access systems are based on division of time, frequency, code or space among different users, leading to time division multiple access (TDMA), frequency division multiple access (FDMA), code division multiple access (CDMA) and space division multiple access (SDMA) methods, respectively.
- (4) Cell-handover – the determination of the geographical location of a mobile phone user and the issue of which cell (or antenna) at any given time should serve a mobile user, as the user roams across different cells, is accomplished from processing the strengths of the radio signals from a mobile user received by different base stations. Usually the base station that has the strongest signal serves the mobile user. The signals from a mobile device received by different base stations can be used to locate the user within a resolution of a few meters or less, depending on the cell size.

- (5) Rake correlators – in a broadband mobile environment, the reflections of an electromagnetic signal, from different surfaces and through different paths, arrive at different times as several distorted replicas of the signal. Rake receivers advantageously use the effect of the multipath propagation by combining the reflections of a signal received from different propagation paths. This can reduce fading and add to the strength of the received signal.
- (6) Channel equalisation is used to remove the distortions and time-dispersion of signals that result from the nonideal characteristics of radio channels. Channel equalisation reduces the symbol overlaps (ISI) and bit error rate at the receiver. It is described in Chapter 15.
- (7) Echo cancellation is used to reduce both acoustic feedback echo between the speaker and microphone of a mobile and the telephone exchange hybrid line echo. Echo cancellation is necessary for voice and data communications systems. It is described in detail in Chapter 14.
- (8) Smart antennas are used for a variety of purposes from increasing the signal-to-noise ratio to space division multiple access. Smart antennas are arrays of phased antennas whose beam direction and gain are controlled by adaptive signal processing methods so that the transmitted electromagnetic power is more efficiently beamed and selectively directed towards the mobile users.

The main method currently used to increase the capacity of radio channels is based on frequency reuse. As explained above, this involves the reuse of the same frequencies in nonadjacent cells where the power of the transmitted electromagnetic wave from a cell fades to insignificance by the time it reaches the nonadjacent cells using the same radio channels. In congested urban areas, the frequency reuse factor can be increased through a reduction of the cell size and transmission power at the expense of more base station infrastructures. In order to minimize the interference among nonadjacent cells, which reuse the same frequencies, the base-stations and the mobile phones operate on low-power transmitters and receivers. Low-power transmitters/receivers have the following advantages:

- (1) low co-channel interference – owing to the low power of the transmissions from base-stations and mobile phones, the electromagnetic waves from each cell fade away before they reach nonadjacent cells that reuse the same frequencies;
- (2) with all mobile devices and base stations operating on low power, the signal-to-noise ratios at the receivers of base stations and mobile phones improve;
- (3) with low transmission power, the power consumption of the cell phones is relatively low – low power consumption implies small batteries with longer talk time and also less exposure to possibly harmful radiations.

17.4 NOISE AND DISTORTION IN MOBILE COMMUNICATIONS SYSTEMS

In classical communications theory it is assumed that the received signal is corrupted by additive white Gaussian noise and the signal distortions due to the channel response are modelled by intersymbol interference. In reality there are many different sources of noise and interference that may limit the performance of a communications system. The most

common sources of distortion and noise in a mobile environment include receiver antenna thermal noise, interference from electromagnetic devices, radiation noise, background noise, echo and, most importantly, multipath and fading, which are described next.

17.4.1 MULTIPATH PROPAGATION OF ELECTROMAGNETIC SIGNALS

There are three main mechanisms that impact the propagation of a radio frequency (RF) electromagnetic wave in the mobile communications environment:

- (1) *Reflection* occurs when an electromagnetic wave impinges on a smooth surface with much larger dimensions than the wavelength, λ , of the radio frequency electromagnetic signal (at the speed of light, $c = 0.3 \times 10^9$ m/s, a radio frequency wave with a frequency $f = 1$ GHz has a wavelength of $\lambda = c/f = 30$ cm).
- (2) *Diffraction* occurs when an electromagnetic wave is obstructed by a dense object with dimensions larger than the wavelength, λ , of the RF electromagnetic signal. The wave then bends and appears as secondary waves from behind the obstructing object. Diffraction and reflection phenomena account for the propagation of electromagnetic waves in cases where there is no line-of-sight radio connection between the transmitter and the receiver
- (3) *Scattering* occurs when a wave impinges on rough objects with large dimensions or from any object with dimensions comparable to or smaller than the wavelength, λ , of the RF electromagnetic signal, causing the wave to scatter in all directions.

In a wireless communications environment the transmitted electromagnetic wave usually encounters a number of different obstacles and reflectors in its propagation path. Hence the transmitted signal and its reflections arrive at the receiver from several different directions over a multiplicity of different paths with each path having a different length and characteristic in terms of fading, phase and time of arrival. This phenomenon is the *multipath effect*. For mobile systems the communications environments change with time, space and the speed of movement of the mobile user and this results in a time-varying multipath channel effect.

A simple illustration of multipath effects in wireless connections is shown in Figure 17.4. The multipath effect is usually described by:

- the line-of-sight path – this is the direct path between the transmitter and the receiver;
- nonline-of-sight paths – the paths arriving after refraction and reflections from various objects and surfaces.

A multipath propagation effect can be modelled as the impulse response of a linear channel as

$$h(t) = \sum_{l=1}^L \alpha_l \delta(t - \tau_l) \quad (17.2)$$

where L is the number of different propagation paths. The multipath Equation (17.2) has two parameters for each propagation path l ; these are the time delay, τ_l , and the amplitude fading factor, α_l .

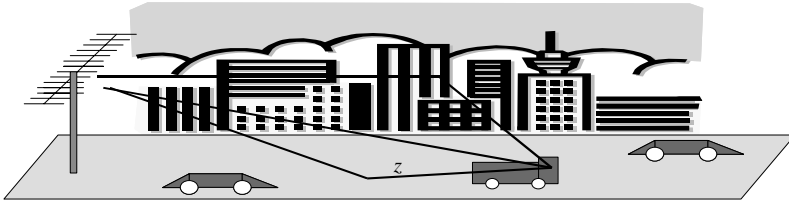


Figure 17.4 A simple illustration of multipath propagation effect in a mobile phone environment.

The multipath effect results in amplitude and phase fluctuations and time delay in the received signals, and this can reduce the capacity and in severe cases create signal outage and loss of connection. The multipath effect is described by the following characteristics:

- *Multipath fading characteristics* – when the reflected waves arriving from different propagation paths are out of phase, a reduction of the signal strength, or fading, at the receiver can occur. There are two types of fading, slow fading due to movements of a mobile user over large areas and fast fading due to movements over smaller distances comparable to the wavelength.
- *Multipath delay spread* – as the multiple reflections of the transmitted signal may arrive at the receiver at different times, this can result in inter-symbol interference and time-dispersion and broadening of the signal. This time dispersion of the channel is called multipath delay spread, which is an important parameter for assessment of the performance of wireless systems.

A common measure of multipath delay spread is the root mean square (rms) delay spread. For reliable communications without using adaptive equalisation or other multipath modelling techniques, the transmitted data rate should be much smaller than the inverse of the rms delay spread, which is called the ‘coherence bandwidth’. When the transmitted data rate is much smaller than the coherent bandwidth, the wireless channel is referred to as the ‘flat channel’ or ‘narrowband channel’. When the transmitted data is closely equal to or larger than the coherent bandwidth, the wireless channel is called the ‘frequency-selective channel’ or ‘wideband channel’.

17.4.2 RAKE RECEIVERS FOR MULTIPATH SIGNALS

The actual form of the signal distortion due to multipath reflections depends on the signal bandwidth, the durations of reflected signals and on the time delays incurred in propagation of a signal via different paths.

The duration of a signal is inversely proportional to its bandwidth. For very wide-band short-duration signals, multipath reflections do not result in intersymbol interference, rather in the appearance of multiple distorted replicas of the transmitted pulses. For example, at a bandwidth of 10 MHz the duration of a pulse is roughly about $0.1 \mu\text{s}$. Now the distance an electromagnetic wave travels in $0.1 \mu\text{s}$ is only 30 m. Hence, any two reflections of pulses travelling distances with differences of more than 30 m would appear as distinct pulses arriving at different times.

Assuming that the noise and fading in different propagation paths are independent, the different versions of a pulse arriving from different paths can be combined, in a Rake receiver, to improve the signal-to-noise ratio. A Rake receiver uses several correlators to individually process several multipath components of a signal. The correlators' outputs are combined to achieve improved SNR and communications reliability and performance. This is a type of so-called 'space diversity gain', i.e. combining the same information arriving from different spatial routes.

Each correlator in a Rake receiver is called a Rake-receiver finger. There are two primary methods used to combine the Rake-receiver finger outputs. One method weights each output equally. The second method uses the data to estimate weights which maximize the SNR of the combined output.

17.4.3 SIGNAL FADING IN MOBILE COMMUNICATIONS SYSTEMS

Modelling the propagation and fading of electromagnetic wave energy in space is important for the calculation of the required transmitter power in mobile communication. In the idealized model of the propagation of an electromagnetic wave in a *free* space there are no obstacles or particles and hence no reflection or loss of energy occurs. The electromagnetic energy radiated by an isotropic source of radio frequency fades in the free space with the square of the distance d as

$$L_s(d) = \left(\frac{4\pi d}{\lambda} \right)^2 = \left(\frac{4\pi d f}{c} \right)^2 \quad (17.3)$$

where L_s is the free space path loss (or power loss) and $\lambda = c/f$ is the wavelength of an electromagnetic wave with frequency f Hz and speed $c = 0.3 \times 10^9$ m/s, which is the speed of light. The fading of the signal strength in a free space is not due to loss of energy, as there is no energy lost in propagation of a signal through a free space, but to the dilution of the energy of the wave, as the same amount of wave energy spreads in propagation through an increasingly larger surface area of a sphere which expands with increasing distance (radius) d .

There are two types of signal fading in mobile communications systems, large-scale *slow fading* due to the movements of the mobile users over large areas and small-scale *fast fading* due to the movements of the mobile users over small distances of the order of the wavelength, λ , of the signal (the wavelength of a mobile carrier at 1 GHz frequency is about $\lambda = c/f = 0.3 \times 10^9 / 10^9 = 30$ cm).

In a mobile system, the received signal, $r(t)$, can be modelled as the convolution of the transmitted signal, $s(t)$, and the impulse response of the radio channel $h_c(t)$ as

$$r(t) = s(t) * h_c(t) \quad (17.4)$$

where $*$ denotes convolution. For mobile phone systems the received signal, $r(t)$, can be further expressed as the product of a slow-fading term, $m(t)$, and a fast-fading term, $r_0(t)$, as

$$r(t) = m(t) \times r_0(t) \quad (17.5)$$

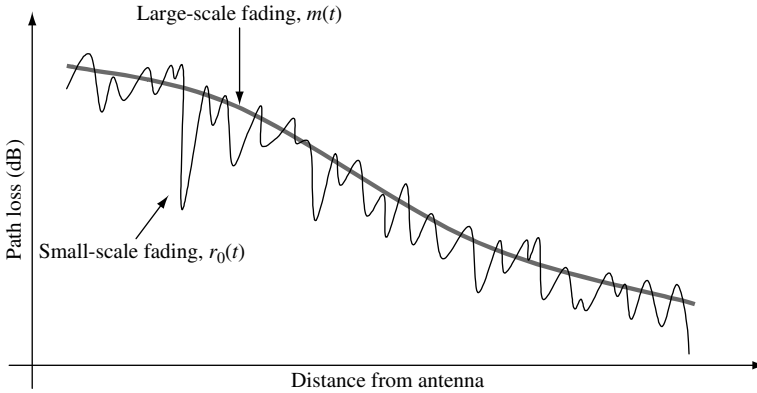


Figure 17.5 Illustration of large-scale and small-scale fading.

Figure 17.5 illustrates the variations of large-scale slow-fading and small-scale fast-fading signals.

17.4.4 LARGE-SCALE SIGNAL FADING

The large-scale path loss due to the movement of a mobile phone user, over a distance d , is obtained from experimental measurements in urban/rural environments. The experimental results for path loss can be modelled with an n th power of distance d as

$$Lp(d) \propto \left(\frac{d}{d_0}\right)^n \quad (17.6)$$

where d_0 is a reference distance which is typically about 1 km for large cells, 100 m for micro cells and 1 m for indoor radio channels. The propagation path loss can be written in terms of a logarithmic dB measure as

$$10 \log L_p(d) = 10 \log L_s(d_0) + 10n \log \left(\frac{d}{d_0}\right) + X_\sigma \text{ dB} \quad (17.7)$$

where $L_s(d_0)$ is the free space loss for the reference distance d_0 , $10n \log(d/d_0)$ is the average path loss in dB as a function of the distance d and X_σ is a random variable that models the random fluctuations of large-scale fading due to random changes in different terrains and environments between the base-stations and mobile users.

The path loss exponent n depends on the propagation environment. In the free space $n = 2$. In some urban street environments with large buildings where a strong waveguide effect may take effect and contain the wave, n can be less than 2 (i.e. less propagation loss than the free space). Generally, where there are obstructions to propagation of the energy of an electromagnetic wave, n is greater than 2.

17.4.5 SMALL-SCALE FAST SIGNAL FADING

Small-scale fading is due to the movements of mobile users over small distances. When a radio wave signal is made from multiple reflective rays and a nonfaded line-of-sight component, then the fluctuations of the amplitude of the signal due to small-scale fading have a distribution that can be modelled by a Rician probability density function, and hence it is known as ‘Rician fading’.

In the absence of a line-of-sight component, the distribution of the amplitude fluctuations of a radio signal caused by small-scale fading has a Rayleigh pdf expressed as

$$p(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & r \geq 0 \\ 0 & r < 0 \end{cases} \quad (17.8)$$

where r is the envelope of the amplitude of the received signal and σ^2 is related to its variance. Often in mobile communications it is assumed that the received signal has a Rayleigh distribution.

The main effects of small-scale fading are:

- (1) Time spreading of the signal pulses – this can result in the received signal having a longer or shorter time span than the transmitted signal. This effect is similar to inter-symbol interference. When the time span of the received signal is greater than the transmitted signal, and there are multiple reflections, the effect can be mitigated by the use of a Rake receiver, as explained earlier. The effect of time spreading can also be mitigated by the use of error control coding and through adding redundancy via diversity gain methods.
- (2) Time-varying behaviour of the communications channel due to spatial movements of the mobile user’s antenna. Relative movement of transmitter and receiver antennas causes a time-varying change in the received amplitude and phase of the signal due to the extra distance that the signal has to travel. Hence, whenever there is spatial movement of the mobile user, the channel becomes time-variant.

17.5 SMART ANTENNAS

Development of more efficient and intelligent antennas is central to the efforts to improve the capacity and reliability of mobile communications systems. An antenna is essentially a wire-to-air impedance-matching device. An antenna transmitter converts an alternating RF electric current signal propagating in a cable or wire to an electromagnetic wave propagating in space, and an antenna receiver has the reverse function. The manner in which electromagnetic energy is distributed onto and collected from the surrounding space has a profound influence on the quality of the reception of radio signals and on the efficient use of the available radio bandwidth and battery power.

Figure 17.6 shows an illustration of the radiation patterns of an omnidirectional antenna and a directional antenna. Omnidirectional antennas radiate electromagnetic energy equally in all directions. Although this is desirable for radio/TV broadcast applications, for personal mobile communications applications it is inefficient and wasteful of power and spectrum because at any given time the mobile user is present at only one place in space and is receiving only a very small fraction of the total radiated power. Furthermore, each user

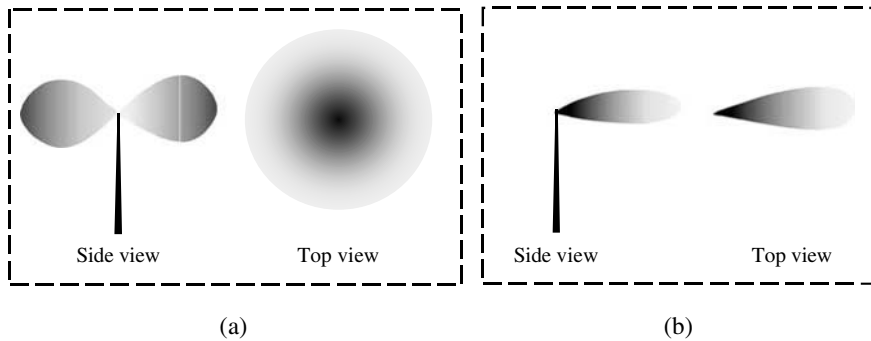


Figure 17.6 Illustration of the radiation pattern of (a) an omnidirectional dipole antenna and (b) a directional antenna.

of a mobile system with an omnidirectional antenna contributes to interference with the communications of other users of mobile systems, who may otherwise be accommodated on the same frequency.

To reduce the inefficiency of omnidirectional antennas, existing mobile base stations divide each cell into a number of fixed sectors with each sector covered by a directional antenna. The conventional practice is to divide each cell into three sectors with each sector covered by an antenna with a 120° beam width.

Instead of using a single antenna for the transmission and reception of signals in a cell or a sector, an array of antennas, and an adaptive signal processing method, can be employed to selectively direct a beam of electromagnetic energy to the mobile user in order to improve the quality of reception and the spectral efficiency.

In general, an array of N antennas can be used in several different ways:

- (1) The output of the transmitter antennas can be adaptively filtered and combined to form a narrow electromagnetic beam directed towards and centred on the mobile user. This arrangement is known as smart antennas and in addition to improving reception and power utilisation it allows a higher level of frequency reuse and hence better spectral efficiency.
- (2) The outputs of the receiver antennas can be added to improve the SNR. Assuming that N antennas receive the same signal with different uncorrelated noise and fading, the output SNR is improved by a factor of N .
- (3) The output of the antenna with the maximum SNR can be used. In its simplest form, with two antennas separated by a half-wavelength, the selection of the output of the antenna with the higher SNR can improve the SNR by more than 10 dB.

Smart beam-forming antennas in their basic form are effectively adaptive phased array antennas with an adaptive signal processing unit that adjusts the phase of the signals that are fed into (or received by) each element of an antenna array in order to achieve a desired beam width and direction. A similar methodology is also used at audio frequency to form adaptive beam-forming directional microphones for speech processing in noise, as explained in Chapter 16.

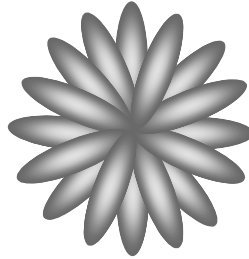


Figure 17.7 Illustration of a switched antenna where the beam is selected from a number of pre-programmed options.

17.5.1 SWITCHED AND ADAPTIVE SMART ANTENNAS

Smart antennas combine inputs from multiple antennas to improve power efficiency and directivity, reduce interference and increase spectral efficiency. There are mainly two forms of smart beam-forming antennas, namely switched antenna array and adaptive antenna array.

As illustrated in Figure 17.7 switched antenna array use a number of pre-selected and programmed RF beams. Switched beam systems combine the outputs of multiple antennas in such a way as to form finely sectorized beams. At any time the signal processing unit estimates the location of the mobile user from its signal strength and then switches the transmitter antenna beam to the one that provides the best signal coverage for the mobile receiver.

Adaptive smart antennas use adaptive signal processing algorithms to continuously adapt the antenna's direction, beam width and gain to follow the movements of the mobile receiver system and to provide optimal reception as the radio environment of the mobile user changes.

17.5.2 SPACE-TIME SIGNAL PROCESSING – DIVERSITY SCHEMES

Space-time signal processing refers to the signal processing methods that utilise the transmission and/or reception of several signals across time and space using multiple transmitter/receiver antennas. The use of MIMO transmitter/receiver antennas together with adaptive signal processing schemes opens up myriad possibilities in the area of space-time signal processing. There are many different ways in which signals from different sources can be arranged, combined and processed for optimal transmission and reception in MIMO mobile systems.

There are three main areas of research and development in the application of antenna arrays to MIMO mobile communications systems:

- (1) the design of the physical antennas as an electromagnetic radiator – this is mainly concerned with such antenna characteristics as the radiation pattern, the main-lobe beamwidth, sidelobe levels and power efficiency;
- (2) the estimation of direction of arrival of the electromagnetic pattern in beam forming antennas using such methods as signal processing methods like MUSIC and ESPRIT, discussed in Chapter 9;

- (3) development of signal processing methods and MIMO coders/decoders for antenna arrays in order to improve spectral efficiency and hence the capacity of radio channels.

An interesting application of space–time processing is the so-called diversity schemes used for robust and efficient transmission in mobile environments.

Diversity schemes deal with the transmission/reception of the replicas of a signal, or a combination of several signals, transmitted via several independent routes, namely time slots, frequency channels, multipath reflections, spatial directions or polarizations. In a diversity scheme all signal routes carry the same combination of messages; however the channel characteristics, noise and fading are independent and uncorrelated. Hence the replicas of the messages from different routes can be processed and combined to increase the SNR. The success of diversity schemes depends on the degree to which the noise and fading on the different diversity branches are uncorrelated and how the information from different routes and channels is processed and combined. Diversity schemes can help to overcome noise and fading in wireless communications channels and increase the channel capacity.

There are a number of diversity schemes described in the literature, and we will briefly consider some basic diversity schemes here:

- *Space diversity* – if the receiver has multiple antennas, the distance between the receiving antennas may be made large enough to ensure independent fading. This arrangement is a form of space diversity. Space separation of half of the wavelength is sufficient to obtain two signals with uncorrelated noise and fading.
- *Polarization diversity* – antennas can transmit either a horizontally polarized wave or a vertically polarized wave. When both waves are transmitted simultaneously, received signals will exhibit uncorrelated fading statistics. This scheme can be considered as a special case of space diversity because separate antennas are used. However, only two diversity branches are available, since there are only two orthogonal polarizations.
- *Angle diversity* – since the received signal arrives at the antenna via several paths, each with a different angle of arrival, directional antennas can isolate the signal component. Each directional antenna will receive a different angular component. Hence, the noise and fading received by different directional antennas pointing at different angles may be uncorrelated.
- *Frequency diversity* – signals with different carrier frequencies far apart from each other could have independent noise and fading. The carrier frequencies must be separated enough so that the fading associated with the different frequencies is uncorrelated. For frequency separations of more than several times the coherence bandwidth, the signal fading would be essentially uncorrelated.
- *Time diversity* – when the same data are sent over the channels at different time instants, the noise and distortions can be uncorrelated if the time separations are large enough and the channels are different. For moving mobile devices time diversity implies that fading and distortions would be uncorrelated as the characteristics of the moving channel change.

17.6 SUMMARY

The reduction of noise, interference, channel distortion and multipath effects is the most important challenge in mobile communications. Cellular architecture and the use of smart

antennas are the main methods for improving the capacity and spectral efficiency in mobile communications systems. This chapter provided an overview of some of the main issues in modelling and reduction of noise and interference in wireless communications systems.

As the demand for multimedia communications on the relatively limited radio spectrum grows, digital array signal processing will play a central part in the development of smart antennas and array noise reduction methods that will take advantage of the opportunities presented by time–space diversity schemes.

BIBLIOGRAPHY

- Balanis C. A., (2005) *Antenna Theory: Analysis and Design*, 3rd edn., John Wiley & Sons, New York.
- Lee Y.W.C. (1989) *Mobile Cellular Communications*. McGraw-Hill, New York.
- Liberti J.C. and Rappaport T.S. (1999) *Smart Antennas for Wireless Communications: IS-95 and Third-Generation CDMA Applications*. Prentice-Hall, Englewood Cliffs, NJ.
- Padgett J.E., Günther C.G. and Hattori T. (1995) Overview of wireless personal communications. *IEEE Commun. Mag.*, **33**(1): 28–41.
- Pahlavan K. and Levesque A.H. (1996) *Wireless Information Networks*. John Wiley & Sons, New York.
- Paulraj A., Nabar R. and Gore D. (2003) *Introduction to Space–Time Signal Processing*. Cambridge University Press, Cambridge.
- Rappaport T.S. (1996) *Wireless Communications: Principles and Practice*, 2nd edn. Prentice-Hall, Englewood Cliffs, NJ.
- Schilling D.L. (1994) Wireless communications going into the 21st century. *IEEE Trans. Vehicular Technol.*, **43**(3): 645–651.
- Shannon C. E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.* **27**: 379–423, 623–656.
- Skelar B. (1997) Rayleigh fading channels in mobile digital communication systems. Part I and II. *IEEE Commun. Mag.* **July**: 90–109.
- Van Trees H.L. (1898) *Detection, Estimation and Modulation Theory, Part I*. John Wiley & Sons, New York.

Index

- Absolute value of error 333
- Acoustic feedback 360
- Acoustic noise 23
- Adaptation formula 196
- Adaptation step size 201, 358
- Adaptive filter 195, 395
- Adaptive noise cancellation 5
- Additive white Gaussian noise 36
- Aliasing 19
- All-pole digital filter 213
- Analogue signals 17
- Autocorrelation 63, 248, 322
- Autocorrelation of (LTI) System 64
- Autocorrelation of impulsive noise 68
- Autocorrelation of white noise 67
- Autocovariance 66
- Autoregressive (AR) model 282, 341
- Autoregressive (AR) process 42, 80, 115, 138, 256
- Autoregressive–moving-average (ARMA) model 254
- AWGN 110

- Backward predictor 219
- Backward probability 148
- Band-limited white noise 25
- Bartlett periodogram 250
- Bayes' rule 48, 146, 228
- Bayesian estimation 103
- Bayesian inference 4
- Bayesian MMSE 107, 418
- Bayesian MMSE spectral amplitude estimation 312
- Bayesian risk function 102

- Beam-forming 13, 427
- Bernoulli–Gaussian model 324
- Bi-cepstrum 403
- Bi-spectrum 401
- Bi-variate pdf 48
- Bias 98
- Binary-state classifier 8
- Binary-state Gaussian process 76
- Blind deconvolution 371
- Blind equalisation 374
- Block least square (BLS) error estimation 171
- Boltzmann constant 30
- Brown noise 26
- Burg's method 224

- Car noise 34
- Cellular communication 434
- Central limit theorem 71, 73, 396
- Channel distortions 24, 32
- Channel equalisation 7
- Channel impulse response 27, 321
- Channel response 371
- Characteristic function 399
- Classification 124
- Clutters 79
- Coherence 70
- Coloured noise 26
- Complete data 116
- Conditional multivariate Gaussian probability 75
- Conditional probability density 49
- Consistent estimator 98

- Continuous density HMM 143, 150
 Continuous-valued random variables 47–8
 Continuously variable state process 138
 Convergence rate 203
 Convolutional noise 396
 Correlation 63, 248, 329
 Correlation-ergodic 72, 170
 Correlation subtraction 234
 Correlator 12
 Cost function 333
 Cost-of-error function 102, 333
 Cramer–Rao lower bound 119
 Cross correlation 68
 Cross-correlation function 346
 Cross covariance 68
 Cross power spectral density 70
 Cumulants 400
 Cumulative distribution function 48
- De-correlation filter 217
 Decision-directed equalisation 392
 Decoding of Signals 152
 Deconvolution 372
 Detection 328
 Detection of signals in noise 12
 Deterministic signals 41
 DFT 310
 Digital coding of audio 10
 Digital signal 17
 Discrete density observation models 149
 Discrete Fourier transform 12, 246
 Discrete state observation HMM 143
 Discrete-time stochastic process 43
 Discrete-valued random variable 46
 Distortion 23
 Distortion matrix 189
 Distortion measurements 430
 Distribution function 73
 Diversity schemes 446
 Divided differences 277
 DNA HMMs 155
 Dolby 15
 Doppler 15
 Durbins algorithm 221
- Echo canceller 356
 Echo suppresser 355
 Echo synthesiser 363
 Efficient estimator 98
 Eigenvalue 202, 203
 Eigenvalue spread 203
 Electromagnetic noise 24, 32
 Electromagnetic radiation 434
 Electrostatic noise 24
 Energy spectral density 248
- Ensemble 43
 Entropy 51, 255
 Entropy, coding 56
 Equalisation 372
 Ergodic HMM 140
 Ergodic processes 43, 70
 ESPRIT algorithm 264
 Estimate–maximise (EM) 116
 Estimation 94
 Estimation of the mean and variance of a Gaussian process 105
 Expected values 62
- Factorisation of linear prediction models 384
 Finite state process 138
 Fisher’s information matrix 121
 Forgetting factor 198
 Formant-tracking 423
 Formant-tracking LP models 230
 Forward predictor model 219
 Forward probability 147
 Fourier series 243
 Fourier transform 245
 Frequency resolution 248
- Gamma pdf 82
 Gaussian (mixture) process 75
 Gaussian pdf 143
 Gaussian process 73
 Gaussian–AR process 115
 Gauss–Markov process 81
- Hard nonlinearity 398
 Hermite polynomials 278
 Hermitian transpose 262
 Hertz 434
 Hidden Markov model 76, 140, 326, 387
 Hidden Markov model for noise 36, 156
 High resolution spectral estimation 259
 Higher-order spectra 401
 HMM-based Wiener filters 415
 HMMs DNA 155
 Homogeneous Poisson process 77
 Homogenous Markov chain 82
 Homomorphic equalisation 380
 Howling 360
 Huber’s function 333
 Huffman code 57
 Hybrid echo 353
 Hypothesised-input HMM equalisation 391
- Ideal equaliser 373
 Ideal interpolation 267
 Impulsive noise 25, 27, 319
 Incomplete data 117

- Influence function 333
 Information 2
 Information models 39, 50
 Inhomogeneous Markov chains 82
 Innovation signal 190, 234
 Inter-symbol interference 393
 Interpolation 267
 Interpolation error 289
 Interpolation through signal substitution 294
 Inverse-channel filter 373
 Inverse filter 216
 Inverse linear predictor 216

 Jacobian 88
 Joint characteristic function 399

 Kalman filter 188, 419
 Kalman gain 190
 K -means clustering algorithm 133
 Kronecker delta function 323

 Lagrange interpolation 274
 Laplacian pdf 83
 Leaky LMS algorithm 205
 Least square AR (LSAR) interpolation 285
 Left–right HMM 141
 Levinson–Durbin algorithm 220, 221
 Line interpolator 275
 Linear array 13
 Linear least squared error filters 165
 Linear prediction models 9, 209, 211, 382
 Linear time invariant channel 180
 Linear transformation 88
 Linear transformation of a Gaussian process 88
 LMS adaptation algorithm 359
 LMS filter 204
 Log–normal process 86

 Magnitude spectral subtraction 301
 Many-to-one mapping 86
 MAP estimation 114
 Marginal density 80
 Marginal probabilities 46, 76
 Markov chain 81
 Markov process 80
 Markovian prior 388
 Markovian state transition prior 142
 M -ary PAM 393
 M -ary pulse amplitude modulation 396
 Matched filter 12
 Matrix inversion lemma 198
 Maximum a posterior (MAP) estimate 103
 Maximum a posterior estimate 228
 Maximum entropy correlation 255
 Maximum phase channel 377, 402

 Maximum phase information 402
 Mean-ergodic 70
 Mean value of a process 63
 Median 109
 Median filters 327
 MIMO 446
 MIMO echo channel identification 366
 Minimisation of backward and forward prediction error 225
 Minimum mean absolute value of error 108
 Minimum mean squared error 169
 Minimum phase channel 376
 Minimum phase information 403
 Mixture Gaussian densities 75, 143
 Mixture pdf 397
 Mobile communication 436
 Model-based signal processing 4
 Model order selection 225
 Model reconstruction 422
 Modelling noise 33, 162
 Monotonic transformation 84
 Moving-average 254
 Multi-input multi-output (MIMO) echo cancellation 365
 Multi-input speech enhancement 425
 Multivariate Gaussian pdf 74
 Multivariate probability mass functions 49
 MUSIC algorithm 261
 Musical noise 305, 307
 Mutual information 54
 M -variate pdf 48

 Narrowband noise 25
 Neural networks 5
 Newton polynomials 276
 Noise 23
 Noise in wireless communication 433
 Noise reduction 177
 Nonlinear spectral subtraction 309
 Nonstationary process 59, 62, 138
 Normal process 73
 Normalised least mean squared error 358
 Nyquist sampling theorem 19, 268

 Observation equation 188
 Outlier 327
 Oversubtraction 308

 Parameter estimation 97
 Parameter space 96
 Parseval’s theorem 176
 Partial correlation 222
 Partial correlation (PARCOR) coefficients 224
 Pattern recognition 8
 Performance measures 98

- Periodogram 249
 Pink noise 27
 Poisson process 77
 Poisson–Gaussian model 324
 Poles and zeros 384
 Posterior pdf 94, 100
 Posterior signal space 94, 101
 Power 60
 Power spectra 380
 Power spectral density 66, 248
 Power spectral subtraction 301
 Power spectrum 176, 242, 250, 322
 Power spectrum estimation 241
 Power spectrum of a white noise 67
 Power spectrum of impulsive noise 67
 Power spectrum subtraction 302
 Prediction error filter 216
 Prediction error signal 217
 Predictive model 94
 Principal eigenvectors 263
 Prior pdf 100
 Prior pdf of the Predictor Coefficients 230
 Prior space of a signal 100
 Probability density function 47
 Probability mass function 45
 Probability models 39, 44
 Processing distortions 304, 307
 Processing noise 24

 QR decomposition 171
 Quantisation 18
 Quantisation noise 21

 Radar 15
 Random signals 41
 Random variable 45
 Rayleigh pdf 83, 444
 Rearrangement matrices 281
 Recursive least squared error (RLS) filter 196
 Reflection coefficient 221
 Robust estimator 333
 Rotation matrix 264

 Sample and hold 18, 19
 Sampling frequency 19
 Scalar Gaussian random variable 73
 Second order statistics 66
 Shannon–Hartley theorem 437
 Short-time Fourier transform (STFT) 292
 Shot noise 78
 Signal 2
 Signal classification 8
 Signal fading 442
 Signal restoration 97
 Signal to impulsive noise ratio 326

 Signal to noise ratio 178
 Signal-to-quantisation noise ratio 21
 Signum nonlinearity 395
 SINR 326
 Sinusoidal signal 42
 Smart antennas 444
 Soft nonlinearity 398
 Source-filter model 211
 Space–time signal processing 446
 Spectral amplitude estimation 297
 Spectral coherence 70
 Spectral efficiency 436
 Spectral subtraction 300, 417
 Spectral whitening 216, 217
 Spectral–time representation 292
 Speech enhancement 409
 Speech processing 9
 Speech recognition 9
 Speech restoration 315
 State-dependent Wiener filters 161
 State equation model 188
 State observation models 143
 State transition probability 142, 149
 State–time diagram 145
 Statistical models 96
 Statistics 41
 Stereophonic echo cancellation 365
 Stochastic processes 43
 Stockham 379
 Strict sense stationary process 61
 Subspace eigenanalysis 259

 Thermal noise 30
 Time-alignment 181
 Time-averaged correlations 170
 Time delay estimation 69
 Time delay of arrival 181
 Time-varying processes 62
 Time/frequency resolutions 247
 Toeplitz matrix 170, 215
 Transform-based coder 11
 Transformation of a random process 83
 Transient noise pulses 25, 29, 337
 Trellis 145
 Tri-cepstrum 404
 Tri-spectrum 401
 Tukey’s bi-weight function 333

 Unbiased estimator 98
 Uncertainty principle 247
 Uniform cost function 103
 Univariate pdf 48

 Vandermonde matrix 275
 Vector quantisation 132, 133

-
- Vector space 172
 - Viterbi decoding algorithm 137, 154
 - Welch power spectrum 251
 - White noise 25, 67
 - Wide sense stationary processes 61
 - Wiener equalisation 377
 - Wiener filter 7, 160, 165, 166, 304, 414
 - Wiener filter in frequency domain 176
 - Wireless communication, noise 433
 - Zero-forcing filter 394
 - Zero-inserted signal 271