FOURTH EDITION

Advanced Digital Signal Processing and Noise Reduction

SAEED V. VASEGHI

WILEY

Advanced Digital Signal Processing and Noise Reduction

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

Advanced Digital Signal Processing and Noise Reduction

Fourth Edition

Professor Saeed V. Vaseghi

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

WILEY

A John Wiley and Sons, Ltd, Publication

This edition first published 2008

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex,
PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Vaseghi, Saeed V.
Advanced digital signal processing and noise reduction / Saeed Vaseghi. — 4th ed.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-470-75406-1 (cloth)
1. Signal processing. 2. Electronic noise. 3. Digital filters (Mathematics)
I. Title.
TK5102.9.V37 2008
621.382′2—dc22
2008027448

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

ISBN 978-0-470-75406-1 (H/B)

Set in 9/11pt Times by Integra Software Services Pvt. Ltd, Pondicherry, India
Printed in Singapore by Markono Print Media Pte Ltd.

To my Luke

Contents

Preface

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 fourth 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 on eigen analysis and principal component analysis and the other on multiple-input multiple-output (MIMO) systems and independent component analysis. In addition the speech enhancement section has been substantially expanded to include bandwidth extension and packet loss replacement.

The applications of DSP are numerous and include multimedia technology, audio signal processing, video signal processing, cellular mobile communication, voice over IP (VoIP), 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.

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 19 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, burst noise, impulsive noise, flicker 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, non-stationary 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 Cramér–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 non-stationary signals. The chapter begins with an introduction to the modelling of non-stationary 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 nonlinear versions of Kalman filter namely extended Kalman and unscented Kalman filters are also considered. The recursive least squared (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 eigen analysis and principal component analysis. Eigen analysis is used in applications such as the diagonalisation of correlation matrices, adaptive filtering, radar signal processing, feature extraction, pattern recognition, signal coding, model order estimation, noise estimation, and separation of mixed biomedical or communication signals. A major application of eigen analysis is in analysis of the covariance matrix of a signal a process known as the principal component analysis (PCA). PCA is widely used for feature extraction and dimension reduction.

Chapter 10 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 non–parametric spectral estimation, model-based spectral estimation, the maximum entropy method, and high–resolution spectral estimation based on eigenanalysis.

Chapter 11 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 squared error interpolation based on an autoregressive model, time–frequency interpolation, and interpolation through search of an adaptive codebook for the best signal.

Chapter 12 considers spectral amplitude estimation. A general form of spectral subtraction is formulated and the processing distortions that result form 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 non-linear methods of spectral subtraction. This chapter also covers the Bayesian minimum mean squared error method of spectral amplitude estimation. This chapter concludes with an implementation of spectral subtraction for signal restoration.

Chapters 13 and 14 cover the modelling, detection and removal of impulsive noise and transient noise pulses. In Chapter 12, impulsive noise is modelled as a binary–state non-stationary 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 15 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 are considered. The chapter concludes with a study of a sub-band echo cancellation system.

Chapter 16 is on 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 17 is on 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 covers the three main areas of noise reduction, bandwidth extension and replacement of missing speech segments. This chapter concludes with microphone array beam-forming for speech enhancement in noise.

Chapter 18 introduces multiple-input multiple-output (MIMO) systems and consider independent component analysis (ICA) for separation of signals in MIMO systems. MIMO signal processing systems are employed in a wide range of applications including multi-sensors biological signal processing systems, phased-array radars, steerable directional antenna arrays for mobile phone systems, microphone arrays for speech enhancement, multichannel audio entertainment systems.

Chapter 19 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 communication systems aim to increase the spectral efficiency defined as the data bits per second per Hz bandwidth of a communication channel. For improved efficiency modern mobile communication 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 communication signal processing methods are employed for improving the speed and capacity of communication systems.

Saeed V. Vaseghi
July 2008

Acknowledgements

I wish to thank Ales Prochazka, Esi Zavarehei, Ben Milner, Qin Yan, Dimitrios Rentzos, Charles Ho and Aimin Chen. Many thanks also to the publishing team at John Wiley, Sarah Hinton, Mark Hammond, Sarah Tilley, and Katharine Unwin.

Symbols

$\boldsymbol{A}$	Matrix of predictor coefficients
a_k	Linear predictor coefficients
$\boldsymbol{a}$	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 $x(m)$
$c_{XX}(k_1, k_2, \cdots, k_N)$	k^{th} order cumulant of $x(m)$
$C_{XX}(\omega_1, \omega_2, \cdots, \omega_{k-1})$	k^{th} order cumulant spectra of $x(m)$
$\boldsymbol{D}$	Diagonal matrix
$e(m)$	Estimation error
$\mathcal{E}[x]$	Expectation of x
f	Frequency variable
Fs	Sampling frequency
$f_X(\boldsymbol{x})$	Probability density function for process $\boldsymbol{X}$
$f_{X,Y}(\boldsymbol{x},\boldsymbol{y})$	Joint probability density function of $\boldsymbol{X}$ and $\boldsymbol{Y}$
$f_{X\|Y}(\boldsymbol{x}\|\boldsymbol{y})$	Probability density function of $\boldsymbol{X}$ conditioned on $\boldsymbol{Y}$
$f_{X;\Theta}(\boldsymbol{x};\boldsymbol{\theta})$	Probability density function of $\boldsymbol{X}$ with $\boldsymbol{\theta}$ as a parameter
$f_{X\|S,\mathcal{M}}(\boldsymbol{x}\|\boldsymbol{s},\mathcal{M})$	Probability density function of $\boldsymbol{X}$ given a state sequence $\boldsymbol{s}$ of an HMM $\mathcal{M}$ of the process $\boldsymbol{X}$
$\boldsymbol{\Phi}(m, m-1)$	State transition matrix in Kalman filter
G	Filter gain factor
$\boldsymbol{h}$	Filter coefficient vector, Channel response
$\boldsymbol{h}_{\max}$	Maximum–phase channel response
$\boldsymbol{h}_{\min}$	Minimum–phase channel response
$\boldsymbol{h}^{\text{inv}}$	Inverse channel response
$H(f)$	Channel frequency response
$H^{\text{inv}}(f)$	Inverse channel frequency response
$\boldsymbol{H}$	Observation matrix, Distortion matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}$	Fisher's information matrix
$\|\boldsymbol{J}\|$	Jacobian of a transformation
$\boldsymbol{K}(m)$	Kalman gain matrix

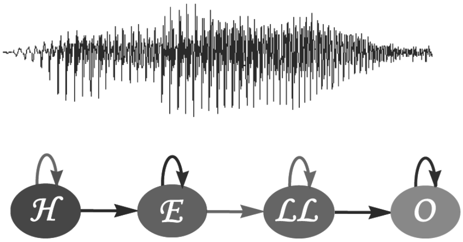
λ	Eigenvalue
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
m	Discrete time index
m_k	k^{th} order moment
$\mathcal{M}$	A model, e.g. an HMM
μ	Adaptation convergence factor
$\boldsymbol{\mu}_{\boldsymbol{x}}$	Expected mean of vector $\boldsymbol{x}$
$n(m)$	Noise
$\boldsymbol{n}(m)$	A noise vector 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
$N\ (\boldsymbol{x}, \boldsymbol{\mu}_{\boldsymbol{xx}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}})$	A Gaussian pdf with mean vector $\boldsymbol{\mu}_{\boldsymbol{xx}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$
$O(\cdot)$	In the order of $(\cdot)$
P	Filter order (length)
$P_X(\boldsymbol{x}_i)$	Probability mass function of $\boldsymbol{x}_i$
$P_{X,Y}(\boldsymbol{x}_i,\ \boldsymbol{y}_j)$	Joint probability mass function of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$
$P_{X\|Y}\left(\boldsymbol{x}_i \middle\| \boldsymbol{y}_j\right)$	Conditional probability mass function of $\boldsymbol{x}_i$ given $\boldsymbol{y}_j$
$P_{NN}(f)$	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 $y(m)$
$\boldsymbol{\theta}$	Parameter vector
$\hat{\boldsymbol{\theta}}$	Estimate of the parameter vector $\boldsymbol{\theta}$
r_k	Reflection coefficients
$r_{xx}(m)$	Autocorrelation function
$\boldsymbol{r}_{\boldsymbol{xx}}(m)$	Autocorrelation vector
$\boldsymbol{R}_{\boldsymbol{xx}}$	Autocorrelation matrix of signal $\boldsymbol{x}(m)$
$\boldsymbol{R}_{\boldsymbol{xy}}$	Cross–correlation matrix
$\boldsymbol{s}$	State sequence
$\boldsymbol{s}^{ML}$	Maximum–likelihood state sequence
σ_n^2	Variance of noise $n(m)$
$\boldsymbol{\Sigma}_{\boldsymbol{nn}}$	Covariance matrix of noise $\boldsymbol{n}(m)$
$\boldsymbol{\Sigma}_{\boldsymbol{xx}}$	Covariance matrix of signal $\boldsymbol{x}(m)$
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
$\boldsymbol{x}(m)$	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 bof the signal $x(m)$
$X(f,t)$	Time-frequency spectrum of the signal $x(m)$
$\boldsymbol{X}$	Clean signal matrix
$\boldsymbol{X}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{X}$
$y(m)$	Noisy signal
$\boldsymbol{y}(m)$	Noisy signal vector
$\hat{y}\,(m\,\|m-i\,)$	Prediction of $\boldsymbol{y}(m)$ based on observations up to time $m-i$
$\boldsymbol{Y}$	Noisy signal matrix

$\boldsymbol{Y}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{Y}$
Var	Variance
w_k	Wiener filter coefficients
$\boldsymbol{w}(m)$	Wiener filter coefficients vector
$W(f)$	Wiener filter frequency response
z	z-transform variable

Abbreviations

AWGN	Additive white Gaussian noise
ARMA	Autoregressive moving average process
AR	Autoregressive process
bps	Bits per second
cdf	Cumulative density function
CELP	Code Excited Linear Prediction
dB	Decibels: 10log10(power ratio)
DFT	Discrete Fourier transform
FIR	Finite impulse response
FFT	Fast Fourier transform
DSP	Digital signal processing
EM	Estimate-maximise
GSM	Global system for mobile
ESPIRIT	Estimation of signal parameters via rotational invariance techniques
GMM	Gaussian mixture model
HMM	Hidden Markov model
Hz	Unit of frequency in cycles per second
ISD	Itakura-Saito distance
IFFT	Inverse fast Fourier transform
IID	Independent identically distributed
IIR	Infinite impulse response
ISI	Inter symbol interference
LMS	Least mean squared error
LP	Linear prediction model
LPSS	Spectral subtraction based on linear prediction model
LS	Least square
LSE	Least square error
LSAR	Least square AR interpolation
LTI	Linear time invariant
MAP	Maximum a posterior estimate
MA	Moving average process
M-ary	Multi-level signalling
MAVE	Minimum absolute value of error estimate
MIMO	Multiple input multiple output
ML	Maximum likelihood estimate

MMSE	Minimum mean squared error estimate
MUSIC	Multiple signal classification
ms	Milliseconds
NLMS	Normalised least mean squared error
pdf	Probability density function
psd	Power spectral density
QRD	Orthogonal matrix decomposition
pmf	Probability mass function
RF	Radio frequency
RLS	Recursive least square
SNR	Signal-to-noise ratio
SINR	Signal-to-impulsive noise ratio
STFT	Short time Fourier transform
SVD	Singular value decomposition
Var	Variance



1

Introduction

Signal processing is concerned with the efficient and accurate modelling, extraction, communication and utilisation of information, patterns and structures in a signal process.

Signal processing provides the theory, the methods and the tools for such purposes as the analysis and modelling of signals, extraction of information from signals, classification and recognition of patterns, synthesis and morphing of signals – morphing is the creation of a new voice or image out of existing samples.

The applications of signal processing methods are very wide and include hi-fi audio, TV and radio, cellular mobile phones, voice recognition, vision, antenna-arrays, radar, sonar, geophysical exploration, medical electronics, bio-medical signal processing, physics and generally any system that is concerned with the communication or processing and retrieval of information. Signal processing plays a central role in the development of new generations of mobile telecommunication and intelligent automation systems and in the efficient transmission, reception, decoding, organisation and retrieval of information content in search engines.

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 biomedical signal processing, adaptive noise reduction, channel equalisation, pattern classification/recognition, audio signal coding, signal detection, spatial processing for directional reception of signals, Dolby noise reduction, radar and watermarking. This chapter concludes with an overview of the most basic process in a digital signal processing system, namely sampling and quantisation.

1.1 Signals, Noise and Information

A signal is the variation of a quantity such as air pressure waves of sounds, colours of an image, depths of a surface, temperature of a body, current/voltage in a conductor or biological system, light, electromagnetic radio waves, commodity prices or volume and mass of an object. A signal conveys information regarding one or more attributes of the source such as the state, the characteristics, the composition, the trajectory, the evolution or the intention of the source. Hence, a signal is a *means of conveying information* regarding the past, the current or the future states of a variable.

For example, astrophysicists analyse the spectrum of signals, the light and other electromagnetic waves, emitted from distant stars or galaxies in order to deduce information about their movements, origins and

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

evolution. Imaging radars calculate the round trip delay of reflected light or radio waves bouncing from the surface of the earth in order to produce maps of the earth.

A signal is rarely observed in isolation from a combination of noise, and distortion. In fact noise and distortion are the fundamental sources of the limitations of: (a) the capacity, or equivalently the maximum speed, to send/receive information in a communication system, (b) the accuracy of measurements in signal processing and control systems and (c) the accuracy of decisions in pattern recognition. As explained in Chapter 2 noise itself is a signal – be it an unwanted signal – that gives information on the state of the source of noise; for example the noise from a mechanical system conveys information on its working order.

A signal may be a function of one dimension, that is a function of one variable, such as speech or music whose amplitude fluctuations are a function of the time variable, or a signal can be multidimensional such as an image (i.e. reflected light intensity) which is a function of two-dimensional space – or a video sequence which is a function of two-dimensional space and time. Note that a photograph effectively projects a view of objects in three-dimensional space onto a two-dimensional image plane where depth information can be deduced from the shadows and gradients of colours.

The information conveyed in a signal may be used by humans or machines (e.g. computers or robots) for communication, forecasting, decision-making, control, geophysical exploration, medical diagnosis, forensics, etc.

The types of signals that signal processing systems deal with include text, image, audio, video, ultrasonic, subsonic, electromagnetic waves, medical, biological, thermal, financial or seismic signals.

Figure 1.1 illustrates a simplified overview of 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 $x(t)$ that carries the information, a communication channel $h[\cdot]$ for modelling the propagation of the signal from the transmitter to the receiver, additive channel and background noise $n(t)$ that exists in every real-life system and a signal processing unit at the receiver for extraction of the information from the received signal.

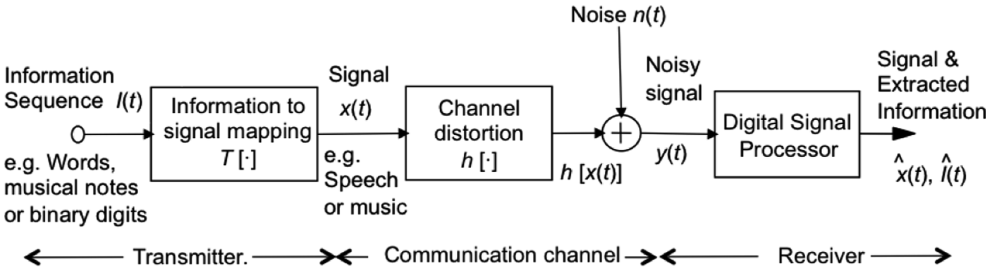

Noise $n(t)$
Information Sequence $I(t)$
e.g. Words, musical notes or binary digits
Information to signal mapping $T[\cdot]$
Signal $x(t)$
e.g. Speech or music
Channel distortion $h[\cdot]$
$h[x(t)]$
Noisy signal
$y(t)$
Digital Signal Processor
Signal & Extracted Information
$\hat{x}(t)$, $\hat{I}(t)$
Transmitter.
Communication channel
Receiver

Figure 1.1 Illustration of a communication and signal processing system.

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

$$x(t) = T[I(t)] \tag{1.1}$$

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 a physical quantity such as voltage or current and modulated to the appropriate form for transmission in a communication channel such as a radio channel, telephone line or cable.

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 room reverberations, or impaired by the speaking abnormalities of the talker, and received as the noisy, distorted and/or incomplete signal $y(t)$ modelled as

$$y(t) = h[x(t)] + n(t) \tag{1.2}$$

Where the function $h[]$ models the channel distortion. In addition to conveying the spoken word, the acoustic speech signal conveys information on the prosody (i.e. pitch intonation and stress patterns) of speech and the speaking characteristic, accent and the emotional state of the talker. The listener extracts this information by processing the signal $y(t)$.

In 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 mobile communication, and to the development of intelligent man/machine interfaces in such areas 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 such as mobile phones, radio and TV.
(2) The extraction of information from noisy and/or incomplete signals for pattern recognition, detection, forecasting, decision-making, signal enhancement, control, automation and search engines.

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

1.2 Signal Processing Methods

Signal processing methods provide a variety of tools for modelling, analysis, coding, synthesis and recognition of signals. Signal processing methods have evolved in algorithmic complexity aiming for the optimal utilisation of the available 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 costs of advanced signal processing methods have been offset and made affordable by the consistent trend in recent years of a continuing increase in 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 are illustrated in Figure 1.2. These methods are described briefly in the following.

1.2.1 Transform-Based Signal Processing

The purpose of a transform is to express a signal or a system in terms of a combination of a set of elementary simple signals (such as sinusoidal signals, eigen vectors or wavelets) that lend themselves to relatively easy analysis, interpretation and manipulation. Transform-based signal processing methods include Fourier transform, Laplace transform, z-transform and wavelet transforms.

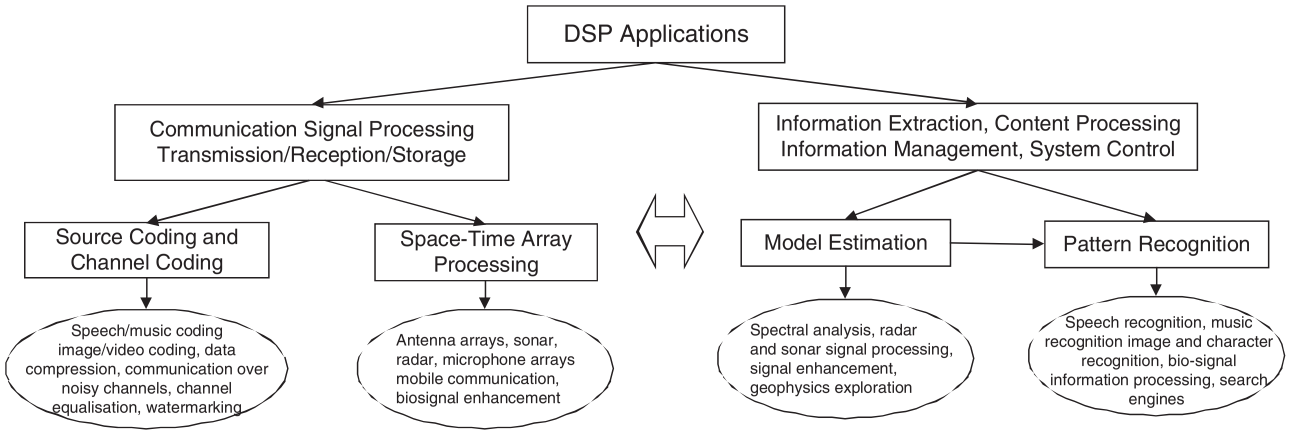


Figure 1.2 A broad categorisation of some of the most commonly used signal processing methods. ICA = Independent Component Analysis, HOS = Higher order statistics. Note that there may be overlap between different methods and also various methods can be combined.

The most widely applied signal transform is the Fourier transform which is effectively a form of vibration analysis; 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 transient sinusoids with exponential amplitude envelopes.

In Fourier, Laplace and z-transform, the different sinusoidal basis functions of each transform all have the same duration and differ in terms of their frequency of vibrations.

In contrast wavelets are multi-resolution transforms in which a signal is described in terms of a combination of elementary waves of different dilations. 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. Wavelet analysis is effectively a tree-structured filter bank analysis in which a set of high pass and low filters are used repeatedly in a binary-tree structure to split the signal progressively into a set of non-uniform sub-bands with different bandwidths.

1.2.2 Source-Filter 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 non-parametric 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 Statistical Model-Based Signal Processing

Statistical signal processing deals with random processes; this includes all information-bearing signals and noise. The fluctuations of a random signal, or the distribution of a class of random signals in the signal space, cannot be entirely modelled by a predictive equation, but it 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 8, 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.

Bayesian inference theory provides a generalised framework for statistical processing of random signals, and for formulating and solving estimation and decision-making problems. Bayesian methods are used for pattern recognition and signal estimation problems in applications such as speech processing, communication, data management and artificial intelligence. In recognising a pattern or estimating a signal, from noisy and/or incomplete observations, Bayesian methods combine the evidence contained in the incomplete signal observation with the prior information regarding the distributions of the signals and/or the distributions of the parameters associated with the signals. Chapter 4 describes Bayesian inference methodology and the estimation of random processes observed in noise.

1.2.4 Neural Networks

Neural networks are combinations of relatively simple non-linear 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 with a hierarchically structured connection network. The connection weights are trained to 'memorise patterns' and perform a signal processing function such as prediction or classification.

Neural networks are particularly useful in non-linear 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 here.

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, telecommunication, 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 DSP applications. This section provides a review of several key applications of digital signal processing methods.

In the following an overview of some applications of DSP is provided. Note that these applications are by no means exhaustive but they represent a useful introduction.

1.3.1 Digital Watermarking

Digital watermarking is the embedding of a signature signal, i.e. the digital watermark, underneath a host image, video or audio signal. Although watermarking may be visible or invisible, the main challenge in digital watermarking is to make the watermark secret and imperceptible (meaning invisible or inaudible). Watermarking takes its name from the watermarking of paper or money for security and authentication purposes.

Watermarking is used in digital media for the following purposes:

(1) Authentication of digital image and audio signals. The watermark may also include owner information, a serial number and other useful information.
(2) Protection of copyright/ownership of image and audio signals from unauthorised copying, use or trade.
(3) Embedding of audio or text signals into image/video signals for subsequent retrieval.
(4) Embedding a secret message into an image or audio signal.

Watermarking has to be robust to intended or unintended degradations and resistant to attempts at rendering it ineffective. In particular watermarking needs to survive the following processes:

(1) Changes in the sampling rate, resolution and format of the signal.
(2) Changes in the orientation of images or phase of the signals.
(3) Noise and channel distortion.
(4) Non-linear imperceptible changes of time/space scales. For example non-linear time-warping of audio or non-linear warping of the dimensions of an image.
(5) Segmentation and cropping of the signals.

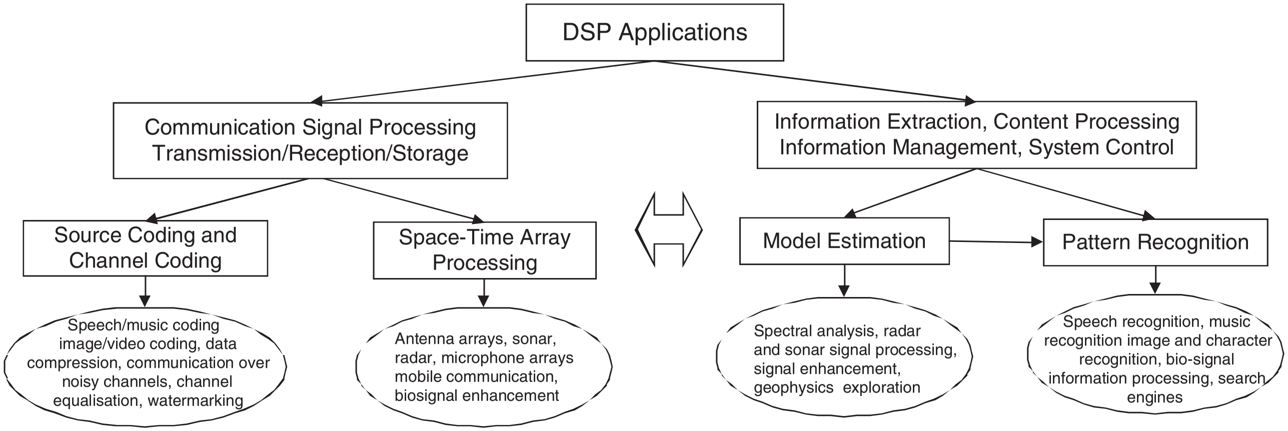


Figure 1.3 A classification of the applications of digital signal processing.

The simplest forms of watermarking methods, illustrated in Figure 1.4, exploit the time–frequency structure of the signal together with the audio-visual perceptual characteristics of humans. The watermark signal is hidden in the parts of the host signal spectrum, where it is invisible in the case of image signals or inaudible in the case of audio signals. Discrete cosine transform or wavelet transform are commonly used for transforming the host signal to frequency-time domains. As shown in Figure 1.4 the watermark is randomised and hidden using a secret key before it is embedded in the host signal. This introduces and additional level of security.

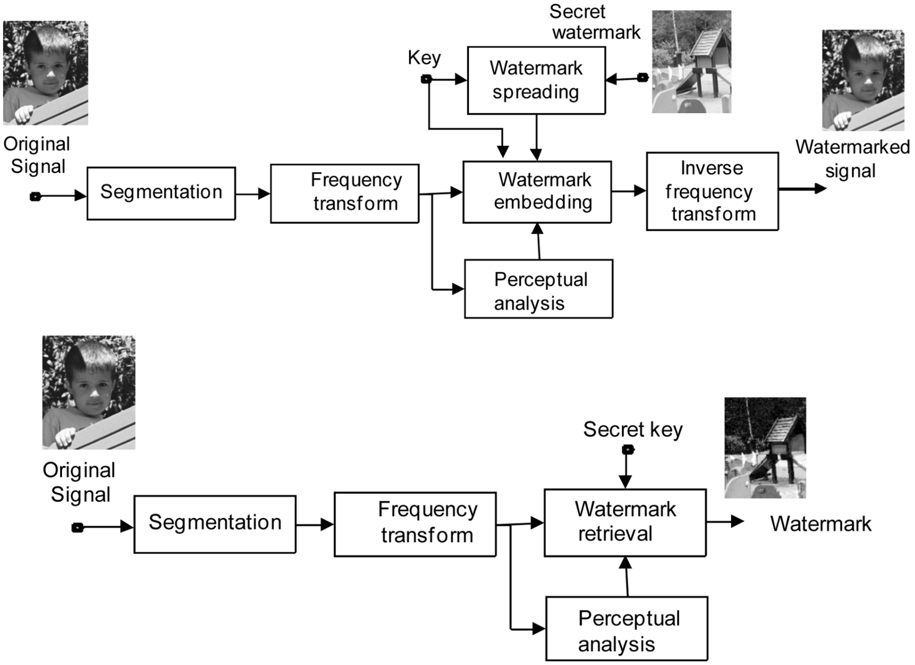


Figure 1.4 A simplified illustration of frequency domain watermark embedding (top) and watermark retrieval (bottom). The secret key introduces an additional level of security. Reproduced by permission of © 2008 Saeed V. Vaseghi.

An example of invisible watermarking is shown in Figure 1.5. The figure shows a host image and another image acting as the watermark together with the watermarked image and the retrieved watermark.

1.3.2 Bio-medical, MIMO, Signal Processing

Bio-medical signal processing is concerned with the analysis, denoising, synthesis and classification of bio-signals such as magnetic resonance images (MRI) of the brain or electrocardiograph (ECG) signals of the heart or electroencephalogram (EEG) signals of brain neurons.

An electrocardiograph signal is produced by recording the electrical voltage signals of the heart. It is the main tool in cardiac electrophysiology, and has a prime function in the screening and diagnosis of cardiovascular diseases.

Electroencephalography is the neurophysiologic measurement of the electrical activity of the neurons in the brain picked up by electrodes placed on the scalp or, in special cases, on the cortex. The resulting

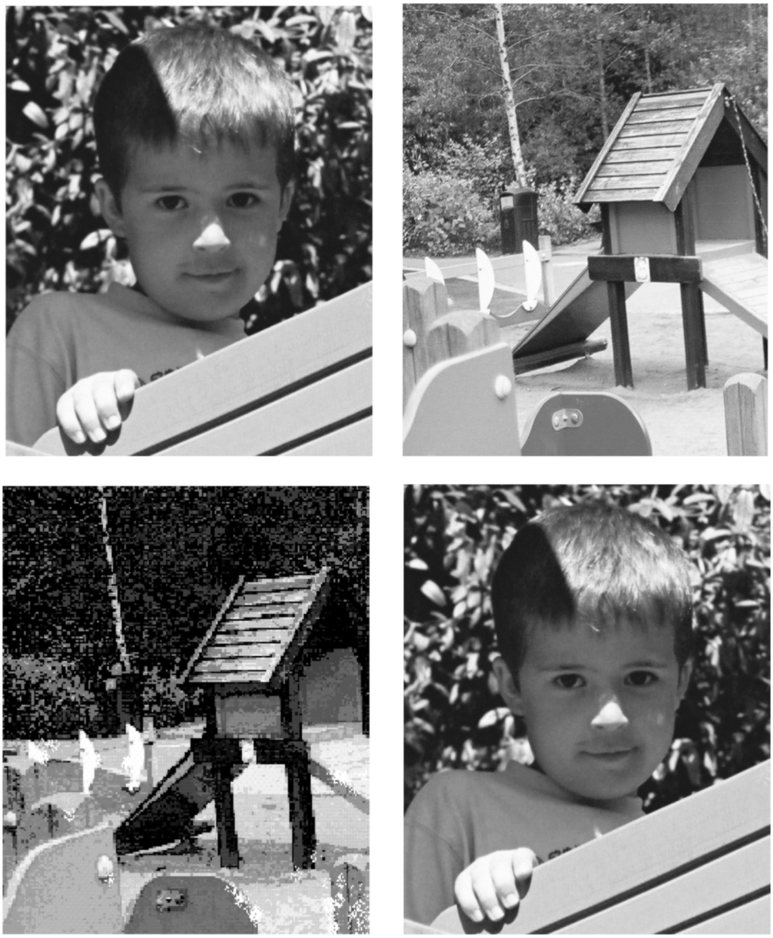


Figure 1.5 Illustration of invisible watermarking of an image, clockwise from top-left: a picture of my son, the watermark, watermarked image and retrieved watermark. The watermark may be damaged due to modifications such as a change of image coding format. Reproduced by permission of © 2008 Saeed V. Vaseghi.

signals are known as an electroencephalograph and represent a mix of electrical signals and noise from a large number of neurons.

The observations of ECG or EEG signals are often a noisy mixture of electrical signals generated from the activities of several different sources from different parts of the body. The main issues in the processing of bio-signals, such as EEG or ECG, are the denoising, separation and identification of the signals from different sources.

An important bio-signal analysis tool, considered in Chapter 18, is known as independent component analysis (ICA). ICA is primarily used for separation of mixed signals in multi-source multi-sensor applications such as in ECG and EEG. ICA is also used for beam forming in multiple-input multiple-output (MIMO) telecommunication.

The ICA problem is formulated as follows. The observed signal vector $\boldsymbol{x}$ is assumed to be a linear mixture of M independent source signals $\boldsymbol{s}$. In a linear matrix form the mixing operation is expressed as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{1.3}$$

The matrix $\boldsymbol{A}$ is known as the *mixing matrix* or the observation matrix. In many practical cases of interest all we have is the sequence of observation vectors $[\boldsymbol{x}(0),\ \boldsymbol{x}(1), \ldots,\ \boldsymbol{x}(N-1)]$. The mixing matrix $\boldsymbol{A}$ is unknown and we wish to estimate a demixing matrix $\boldsymbol{W}$ to obtain an estimate of the original signal $\boldsymbol{s}$.

This problem is known as *blind source separation* (BSS); the term blind refers to the fact that we have no other information than the observation $\boldsymbol{x}$ and an assumption that the source signals are independent of each other. The demixing problem is the estimation of a matrix $\boldsymbol{W}$ such that

$$\hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} \tag{1.4}$$

The details of the derivation of the demixing matrix are discussed in Chapter 18 on ICA. Figure 1.6 shows an example of ECG signal mixture of the hearts of a pregnant mother and foetus plus other noise and interference. Note that application of ICA results in separation of the mother and foetus heartbeats. Also note that the foetus heartbeat rate is about 25 % faster than the mother's heart-beat rate.

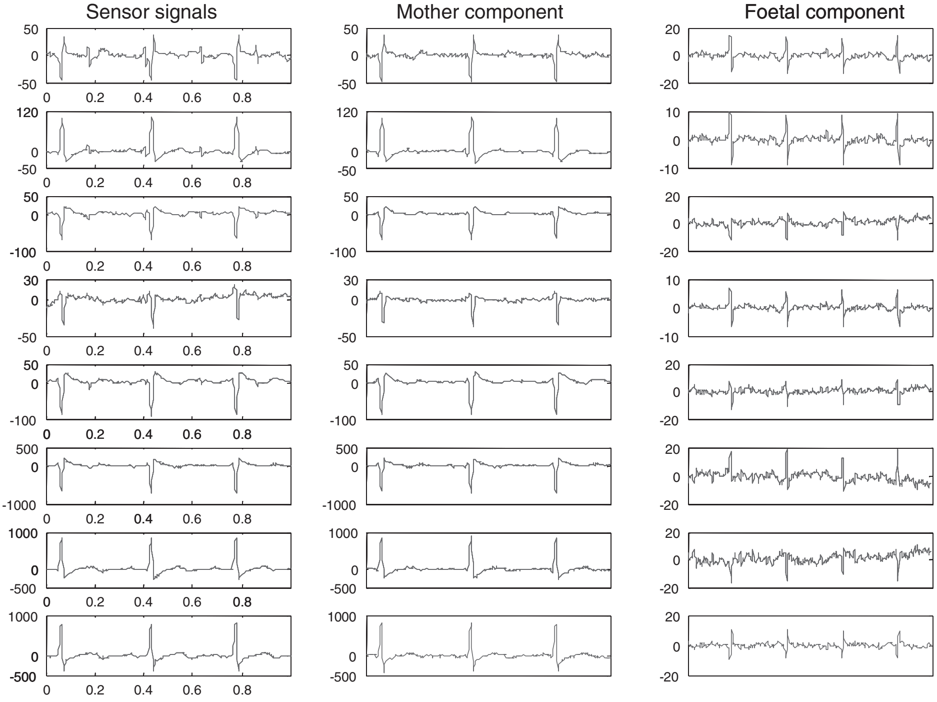


Figure 1.6 Application of ICA to separation of mother and foetus ECG. Note that signals from eight sensors are used in this example.

1.3.3 Echo Cancellation

Echo is the repetition of a signal back to the transmitter; either due to a coupling between the loudspeaker and microphone or due to a reflection of the transmitted signal from the points or surfaces where the characteristics of the medium through which the signal propagates changes significantly so as to impede the propagation of the signal in the original direction such that some of the signal energy is reflected back to the source.

Modern telecommunication systems, Figure 1.7, connect a variety of voice-enabled terminals, such as fixed telephones, mobile phones, laptops etc. via a variety of networks and relays including public switched telephone network (PSTN), satellites, cellular networks, voice over internet protocol (VoIP), wifi, etc. Echo can severely affect the quality and intelligibility of voice conversation in telephone, teleconference, VoIP or cabin communication systems.

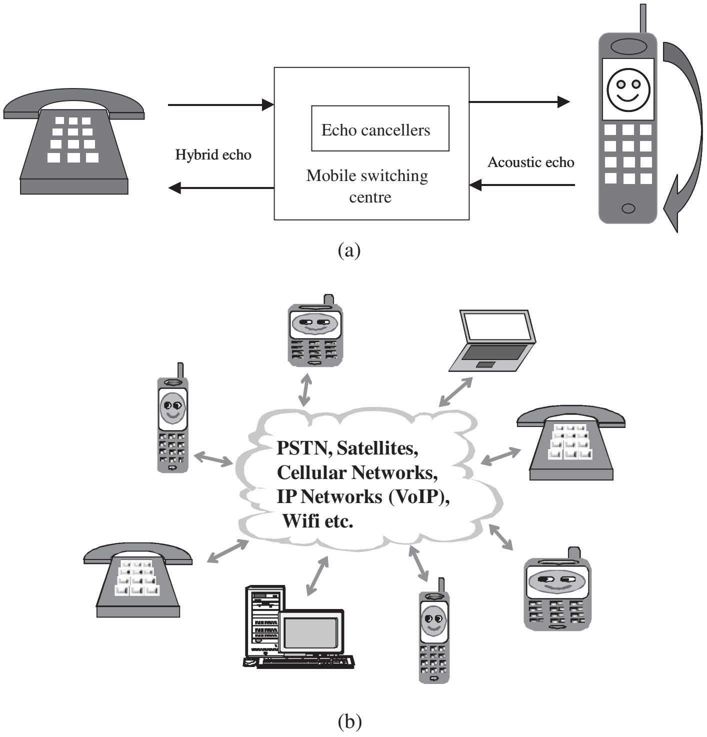


Figure 1.7 (a) Illustration of sources of echo in a mobile-to-landline system, (b) a modern communication network connects a variety of voice-enabled devices through a host of different telephone and IP networks.

Echo cancellation is an important aspect of the design of telecommunication systems such as conventional wire-line telephones, hands-free phones, cellular mobile (wireless) phones, teleconference systems, voice over internet (VoIP) and in-vehicle cabin communication systems.

There are two types of echo in a voice communication system (Figure 1.7(a)):

(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.

Voice communication systems cannot function properly without echo cancellation systems. A solution used in the early days was echo suppression. However, modern communication systems employ adaptive echo cancellation systems that identify the echo path and synthesis a replica of the echo that is subtracted from the actual echo in order to remove the echo. Echo cancellation is covered in Chapter 15.

1.3.4 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. 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

$$y(m) = x(m) + n(m) \tag{1.5}$$

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 1.8 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 non-stationary 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.

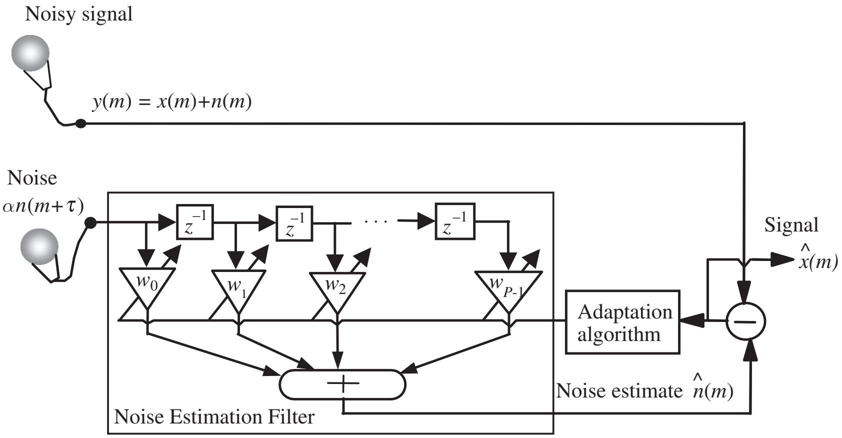


Figure 1.8 Configuration of a two-microphone adaptive noise canceller. The adaptive filter delay elements (z^{-1}) and weights w_i model the delay and attenuation that signals undergo while propagating in a medium.

1.3.5 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 average sense, using the statistics of the signal and the noise process. Figure 1.9 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

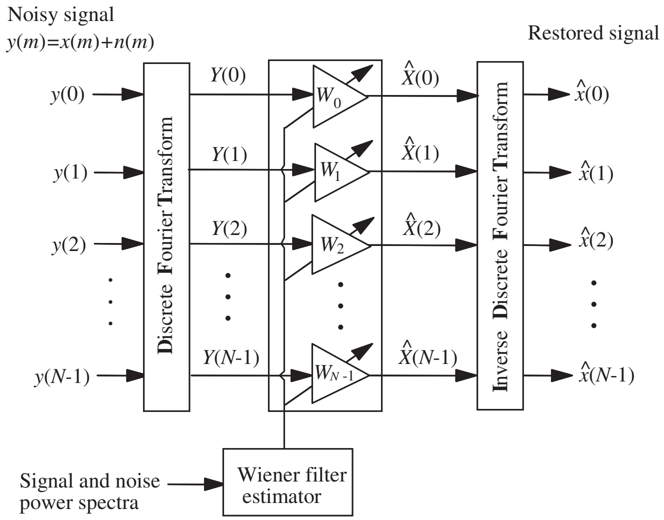


Figure 1.9 A frequency–domain Wiener filter for reducing additive noise.

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.

1.3.6 Blind Channel Equalisation

Channel equalisation is the recovery of a signal distorted in transmission through a communication channel with a non-flat magnitude and/or a non-linear 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 non-ideal channel and multi-path propagation, in speech recognition for removal of the effects of the microphones and the communication channels, in correction of distorted images, analysis of seismic data, de-reverberation of acoustic gramophone recordings etc.

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 1.10 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 non-linear 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 signal *to direct* the equaliser adaptation process. Blind equalisation is covered in detail in Chapter 16.

1.3.7 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.

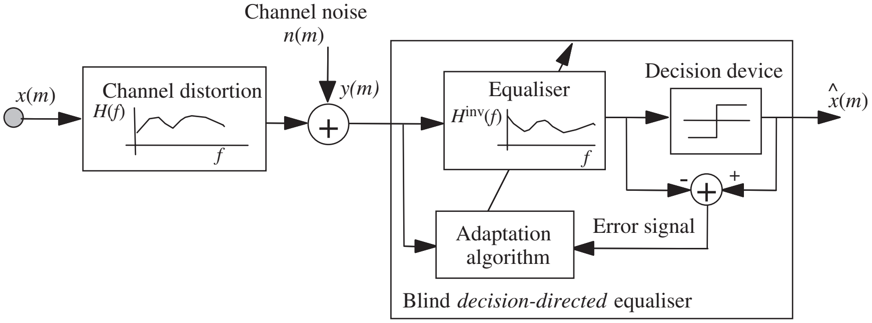


Figure 1.10 Configuration of a decision-directed blind channel equaliser.

To design a classifier, a set of models are 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 codebook, 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 1.11 illustrates a correlation receiver for a BPSK signalling 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.

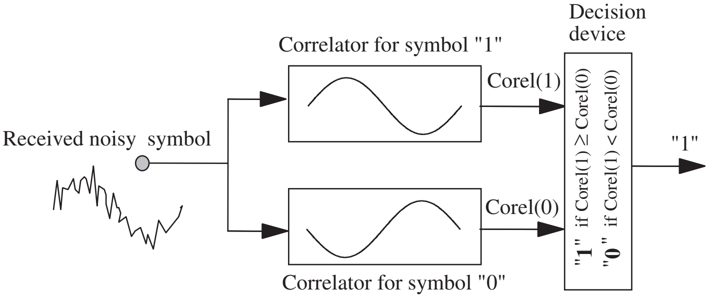

Decision device
Correlator for symbol "1"
Corel(1)
Received noisy symbol
"1" if Corel(1) ≥ Corel(0)
"0" if Corel(1) < Corel(0)
"1"
Corel(0)
Correlator for symbol "0"

Figure 1.11 Block diagram illustration of the classifier in a binary phase-shift keying demodulation.

Figure 1.12 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.

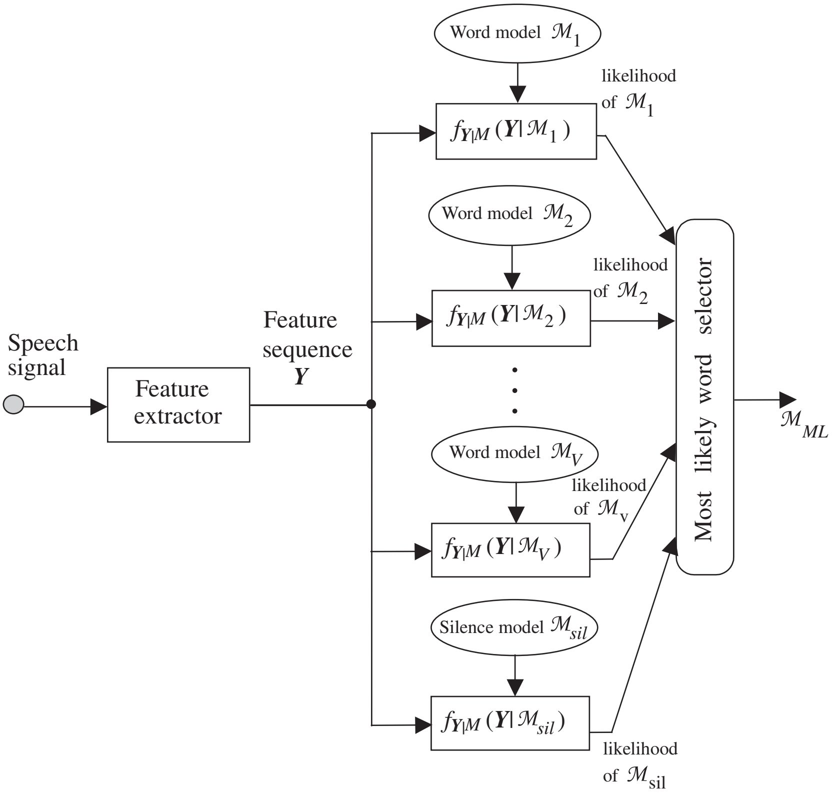

Word model $\mathcal{M}_1$
likelihood of $\mathcal{M}_1$
$f_{Y|M}(\boldsymbol{Y}|\mathcal{M}_1)$
Word model $\mathcal{M}_2$
likelihood of $\mathcal{M}_2$
$f_{Y|M}(\boldsymbol{Y}|\mathcal{M}_2)$
Speech signal
Feature extractor
Feature sequence $\boldsymbol{Y}$
Word model $\mathcal{M}_V$
likelihood of $\mathcal{M}_v$
$f_{Y|M}(\boldsymbol{Y}|\mathcal{M}_V)$
Silence model $\mathcal{M}_{sil}$
$f_{Y|M}(\boldsymbol{Y}|\mathcal{M}_{sil})$
likelihood of $\mathcal{M}_{sil}$
Most likely word selector
$\mathcal{M}_{ML}$

Figure 1.12 Configuration of speech recognition system, $f_{Y|\mathcal{M}}(\boldsymbol{Y}|\mathcal{M}_i)$ is the likelihood of the model $\mathcal{M}_i$ given an observation sequence $\boldsymbol{Y}$.

1.3.8 Linear Prediction Modelling of Speech

Linear predictive models (introduced in Chapter 8) 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 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 resonance of the vocal tract is to shape the frequency spectrum of speech and introduce a measure of correlation and predictability on the random variations of the air from the lungs. Figure 1.13 illustrates a source-filter model for speech production. The source models the lungs

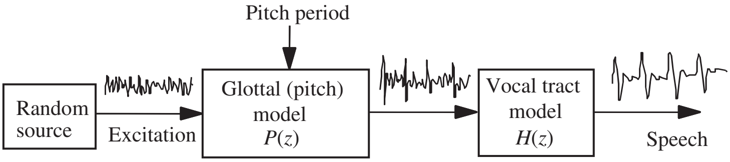


Figure 1.13 Linear predictive model of speech.

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 is an adaptive filter that forecasts the amplitude of the signal at time m, $x(m)$, using a linear combination of P previous samples $[x(m-1), \cdots, x(m-P)]$ as

$$\hat{x}(m) = \sum_{k=1}^{P} a_k x(m-k) \tag{1.6}$$

where $\hat{x}(m)$ is the prediction of the signal $x(m)$, and the vector $\boldsymbol{a}^{\mathrm{T}} = [a_1, \ldots, a_P]$ 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

$$e(m) = x(m) - \sum_{k=1}^{P} a_k x(m-k) \tag{1.7}$$

In speech processing, the prediction error $e(m)$ may also be interpreted as the random excitation or the so-called innovation content of $x(m)$. From Equation (1.7) a signal generated by a linear predictor can be synthesised as

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + e(m) \tag{1.8}$$

Linear prediction models can also be used in a wide range of applications to model the correlation or the movements of a signal such as the movements of scenes in successive frames of video.

1.3.9 Digital Coding of Audio Signals

In digital audio, the memory required to record a signal, the bandwidth and power 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 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 1.14 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 lungs, and the filter models the vibrations of the glottal cords and the vocal tract. At the transmitter, speech is segmented into blocks of 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 1.15. The aim of transformation is to convert the signal into a form where it lends itself to a more convenient and useful interpretation and manipulation. In Figure 1.15 the input signal may be transformed to the frequency domain using a discrete Fourier

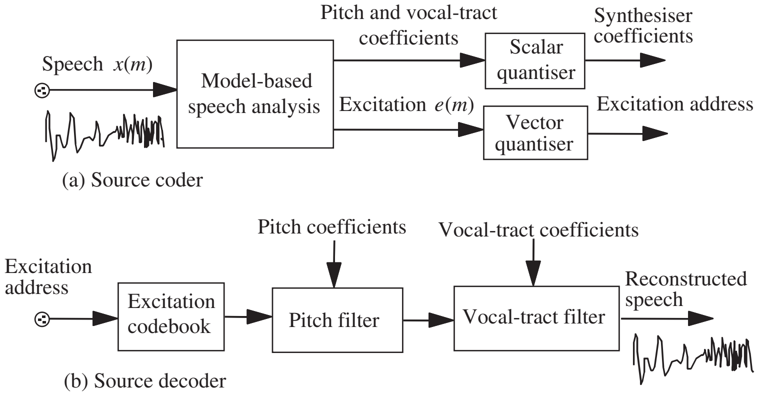


Figure 1.14 Block diagram configuration of a model-based speech (a) coder and (b) decoder.

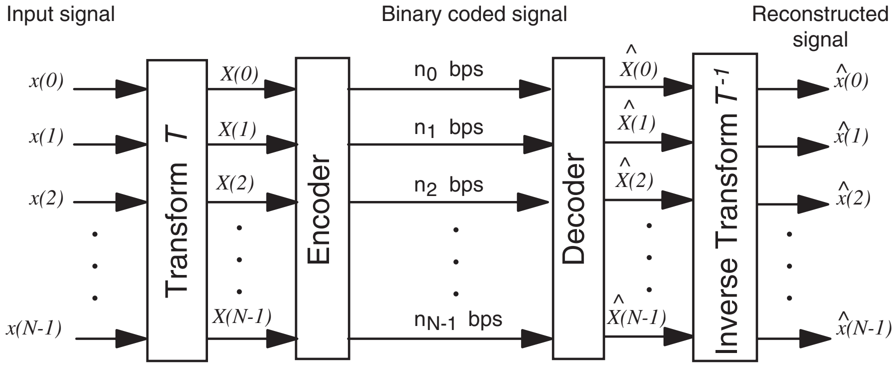


Figure 1.15 Illustration of a transform-based coder.

transform or a discrete cosine transform or a filter bank. 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.

1.3.10 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

$$y(m) = b(m)x(m) + n(m) \tag{1.9}$$

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 1.16. 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

$$h(m) = x(N - 1 - m) \qquad 0 \le m \le N - 1 \tag{1.10}$$

where N is the length of. The output of the matched filter is given by

$$z(m) = \sum_{k=0}^{N-1} h(k)y(m - k) \tag{1.11}$$

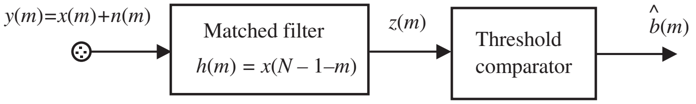


Figure 1.16 Configuration of a matched filter followed by a threshold comparator for detection of signals in noise.

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

$$\hat{b}(m) = \begin{cases} 1 & \text{if } \operatorname{abs}\big(z(m)\big) \ge \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

where $\hat{b}(m)$ is an estimate of the binary state indicator sequence $b(m)$, and it may be erroneous in particular 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 less the likelihood that noise would be classified as signal, 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

$$\mathcal{R}(\text{Threshold} = \theta) = P_{\text{FalseAlarm}}(\theta) + P_{\text{Miss}}(\theta) \tag{1.13}$$

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

Table 1.1 Four possible outcomes in a signal detection problem.

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

1.3.11 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 1.17) to steer the reception of signals towards a desired direction, and in speech processing for reducing the effects of ambient noise.

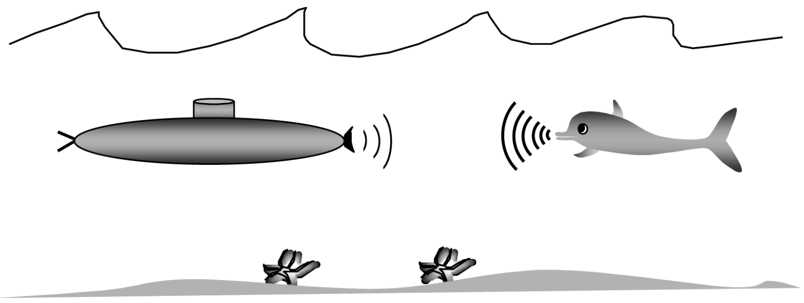


Figure 1.17 Sonar: detection of objects using the intensity and time delay of reflected sound waves.

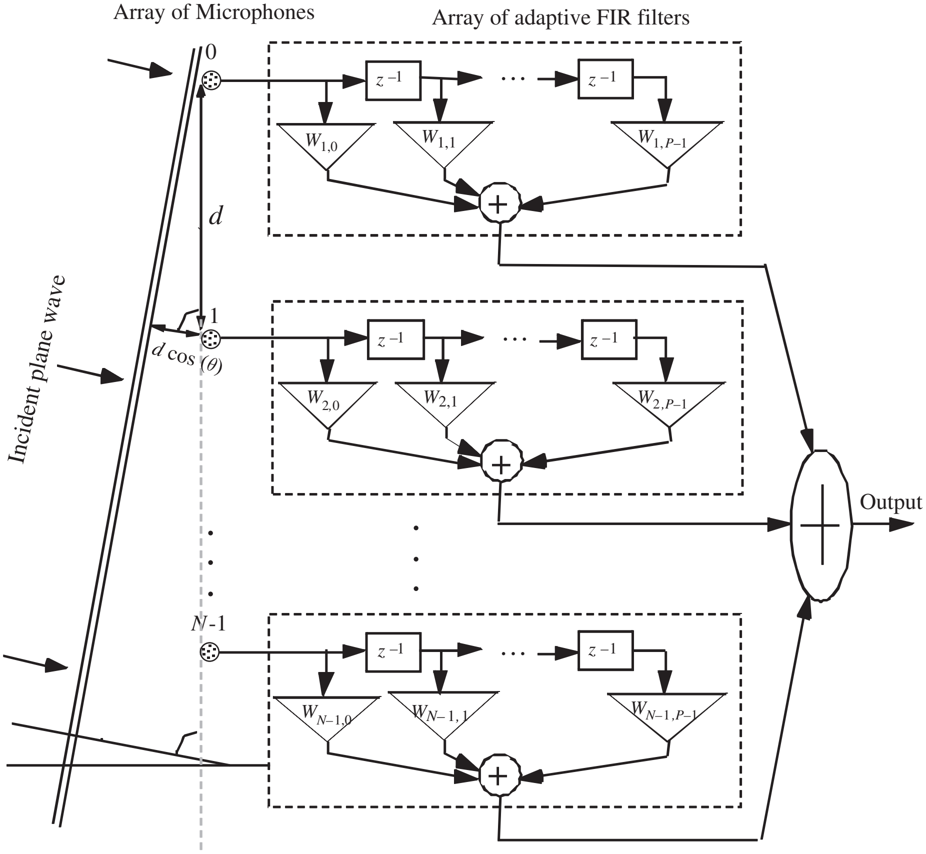


Figure 1.18 Illustration of a beam-former, for directional reception of signals.

To explain the process of beam-forming, consider a uniform linear array of sensors as illustrated in Figure 1.18. 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 1.18.

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

$$\tau = \frac{dcos(\theta)}{c} \tag{1.14}$$

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

$$\phi = 2\pi \frac{\tau}{T_0} = 2\pi F_0 \frac{d\cos\theta}{c} \tag{1.15}$$

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 the direction θ will be time-aligned and coherently combined, whereas those arriving from other directions will suffer cancellations and attenuations. Figure 1.18 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 towards a desired direction, for example towards 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.

1.3.12 Space-Time Signal Processing

Conventionally transmission resources are shared among subscribers of communication systems through the division of time and frequency leading to such resource-sharing schemes as time division multiple access or frequency division multiple access. Space provides a valuable additional resource that can be used to improve both the communication capacity and quality for wireless communication systems.

Space-time signal processing refers to signal processing methods that utilise simultaneous transmission and reception of signals through multiple spatial routes. The signals may arrive at the destinations at different times or may use different time slots. Space-time signal processing, and in particular the division of space among different users, is an important area of research and development for improving the system capacity in the new generations of high-speed broadband multimedia mobile communication systems.

For example, in mobile communication the multi-path effect, where a radio signal propagates from the transmitter to the receiver via a number of different paths, can be used to advantage in space-time signal processing. The multiple noisy versions of a signal, arriving via different routes with different noise and distortions, are processed and combined such that the signal components add up constructively and become stronger compared to the random uncorrelated noise. The uncorrelated fading that the signals suffer in their propagation through different routes can also be mitigated.

The use of transmitter/receiver antenna arrays for beam-forming allows the division of the space into narrow sectors such that the same frequencies, in different narrow spatial sectors, can be used for simultaneous communication by different subscribers and/or different spatial sectors can be used to transmit the same information in order to achieve robustness to fading and interference. In fact combination of space and time can provide a myriad of possibilities, as discussed in Chapter 19 on mobile communication signal processing. Note that the ICA method, described in Section 1.3.2 and Chapter 18, is often used in space-time signal processing for separation of multiple signals at the receiver.

1.3.13 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 even by a low 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 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 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 Hz 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 of each band is adaptively adjusted to boost low-energy signal components. Dolby A provides a maximum gain of 10 to 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 than the maximum reference) and Dolby C provides a boost of up to 20 dB as illustrated in Figure 1.19.

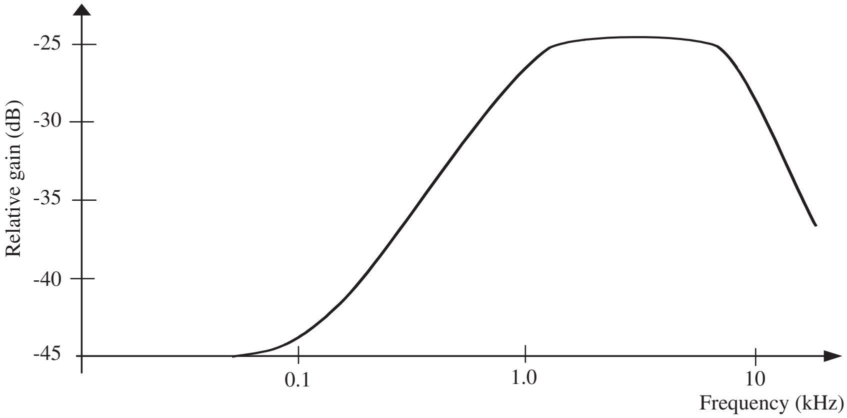


Figure 1.19 Illustration of the pre-emphasis response of Dolby C: up to 20 dB boost is provided when the signal falls 45 dB below maximum recording level.

1.3.14 Radar Signal Processing: Doppler Frequency Shift

Figure 1.20 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 a flying aeroplane. A radar system consists of a 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 received echo provides such information as range, speed and acceleration. The received signal has the form

$$x(t) = A(t)\cos\{\omega_0[t - 2r(t)/c]\} \tag{1.16}$$

where $A(t)$, the time-varying amplitude of the reflected wave, depends on the position and the 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

$$r(t) = r_0 + \dot{r}t + \frac{1}{2!}\ddot{r}t^2 + \frac{1}{3!}\dddot{r}\,t^3 + \cdots \tag{1.17}$$

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

$$r(t) \approx r_0 + \dot{r}t \tag{1.18}$$

Substituting Equation (1.18) in Equation (1.16) yields

$$x(t) = A(t)\cos[(\omega_0 - 2\dot{r}\omega_0/c)t - 2\omega_0 r_0/c] \tag{1.19}$$

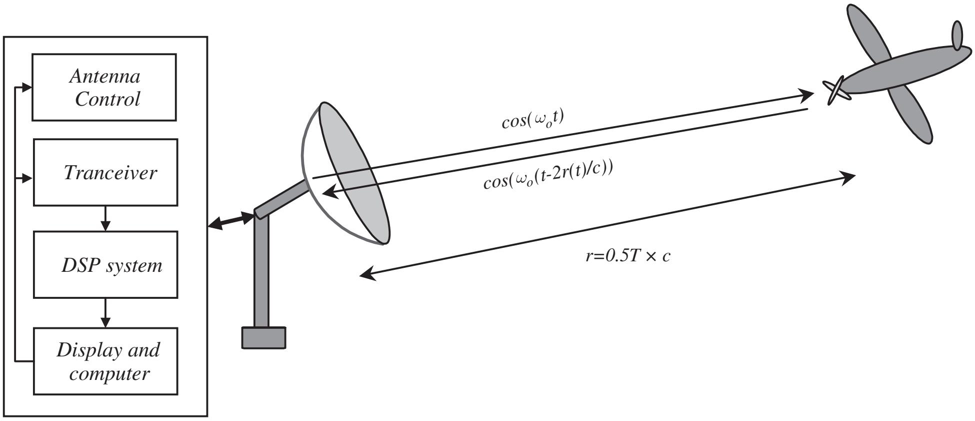


Figure 1.20 Illustration of a radar system.

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

$$\omega_d = 2\dot{r}\omega_0/c \tag{1.20}$$

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

$$r_0 = 0.5\ T \times c \tag{1.21}$$

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 A Review of Sampling and Quantisation

Digital signal processing involves the processing of signals by a computer or by a purpose-built signal processing microchip. The signal is stored in the computer's memory in a binary format in terms of a sequence of n-bit words. Hence, to digitally process signals that are not already in a digital format, the signals need to be converted into a digital format that can be stored and processed in a computing device.

Sampling and quantisation are the first two steps in all digital signal processing and digital communication systems which have analogue inputs. Most signals such as speech, image and electromagnetic waves, are not naturally in a digital format but need to be digitised (i.e. sampled and quantised) for subsequent processing and storage in a digital system such as in a computer or in a mobile DSP chip or a digital music player.

A signal needs to be sampled at a rate of more than twice the highest frequency content of the signal; otherwise the sampling process will result in loss of information and distortion. Hence, prior to sampling, the input signal needs to be filtered by an anti-aliasing filter to remove the unwanted signal frequencies above a preset value of less than half the sampling frequency. Each sample value is subsequently quantised to the nearest of 2^n quantisation levels and coded with an n-bit word.

The digitisation process should be performed such that the original continuous signal can be recovered from its digital version with no loss of information, and with as high a fidelity as is required in an application.

A digital signal is a sequence of discrete real-valued or complex-valued numbers, representing the fluctuations of an information-bearing quantity with time, space or some other variable.

The most elementary unit of a discrete-time (or discrete-space) signal is the unit-sample signal $\delta(m)$ defined as

$$\delta(m) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \tag{1.22}$$

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

$$x(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(m-k) \tag{1.23}$$

Figure 1.21 illustrates a discrete-time signal and its continuous-time envelope. Many signals such as speech, music, image, video, radar, sonar and bio-signals and medical signals are analogue, in that in their original form they appear to vary continuously with time (and/or space) and are sensed by analogue sensors such as microphones, optical devices and antennas. Other signals such as stock market prices are inherently discrete-time and/or discrete amplitude signals. Continuous signals are termed analogue signals because their fluctuations with time are analogous to the variations of the signal source.

For digital processing of continuous signals, the signals are first sampled and then each sample is converted into an n-bit binary digit. The sampling and digitisation process should be performed such that the original continuous signal can be recovered from its digital version with no loss of information, and with as high a fidelity as is required in an application.

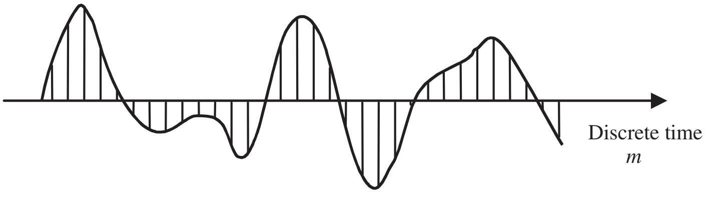


Figure 1.21 A discrete-time signal and its continuous-time envelope of variation.

Analogue-to-digital conversion, that is the conversion of an analogue signal into a sequence of n-bit words, consists of the two basic steps of sampling and quantisation:

(1) *Sampling*. The first step is to sample a signal to produce a discrete-time and/or discrete-space signal. The sampling process, when performed with sufficiently high frequency (greater than twice the highest frequency), can capture the fastest fluctuations of the signal, and can be a loss-less operation in that the original analogue signal can be recovered through interpolation of the sampled sequence.

(2) *Quantisation*. The second step is quantisation of each sample value into an n-bit word. Quantisation involves some irrevocable errors and possible loss of information. However, in practice the quantisation error (aka quantisation noise) can be made negligible by using an appropriately high number of bits as in a digital audio hi-fi.

Figure 1.22 illustrates a block diagram configuration of a digital signal processor with an analogue input. The anti-aliasing low-pass filter (LPF) removes the out-of-band signal frequencies above a pre-selected cut-off frequency which should be set to less than half the intended sampling frequency. The sample-and-hold (S/H) unit periodically samples the signal to convert the continuous-time signal into a discrete-time, continuous-amplitude signal.

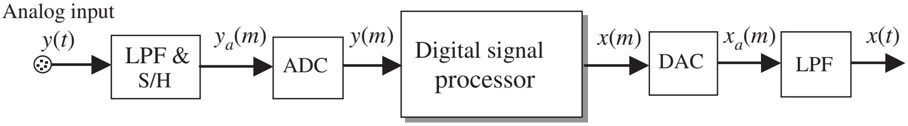


Figure 1.22 Configuration of a digital signal processing system with analogue input and output.

The analogue-to-digital converter (ADC) follows the S/H unit and maps each continuous-amplitude sample into an n-bit word. After the signal is processed, 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 1.22. Figure 1.23 shows a sample-and-hold circuit diagram where a transistor switch is turned ‘on’ and ‘off’ thereby allowing the capacitor to charge up or down to the level of the input signal during the ‘on’ periods and then holding the sample value during the ‘off’ period.

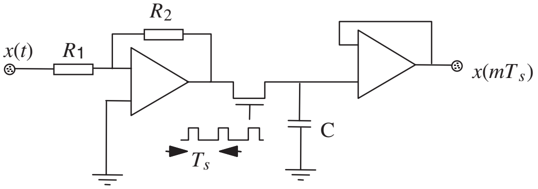


Figure 1.23 A simplified sample-and-hold circuit diagram; when the switch closes the capacitor charges or discharges to the input level.

1.4.1 Advantages of Digital Format

The advantages of the digital format are as follows:

(1) *Digital devices such as mobile phones are pervasive*.

(2) *Transmission bandwidth and storage space savings*. Digital compression techniques, such as MP3, can be used to compress a digital signal. When combined with error-control coding and efficient

digital modulation methods the required overall bandwidth is less than that of say an FM-modulated analogue signal of similar noise robustness and quality. There is a similar reduction in storage requirement.

(3) *Power savings*. Power saving depends on the compression rate and the modulation method. In general digital systems can achieve power efficiency compared with analogue systems.

(4) *Noise robustness*. Digital waveforms are inherently robust to noise and additional robustness can be provided through error-control coding methods.

(5) *Security*. Digital systems can be encrypted for security, and in particular the code division multiple access (CDMA) method, employed for sharing of time/bandwidth resources in mobile phone networks, is inherently secure.

(6) *Recovery and restoration*. Digital signals are more amenable to recovery of lost segments.

(7) *Noise reduction*. Digital noise reduction methods can be used to substantially reduce noise and interference and hence improve the perceived quality and intelligibility of a signal.

(8) *Editing and mixing* of audio/video and other signals in digital format is relatively easy.

(9) *Internet and multimedia systems*. Digital communication, pattern recognition, Internet and multimedia communication would not have been possible without the digital format.

1.4.2 Digital Signals Stored and Transmitted in Analogue Format

Digital signals are actually stored and transmitted in analogue format. For example, a binary-state transistor stores a one or a zero as a quantity of electronic charge, in bipolar baseband signalling a ‘1’ or a ‘0’ is signalled with a pulse of $\pm V$ volts and in digital radio-frequency signalling binary digits are converted to modulated sinusoidal carriers for transmission over the airwaves. Also the digital data on a CD track consists of a sequence of bumps of micro to nanometre size arranged as a single, continuous, long spiral track of data.

1.4.3 The Effect of Digitisation on Signal Bandwidth

In its simplest form each binary bit (‘1’ or ‘0’) in a bit-stream representation of a signal can be viewed as pulse of duration T seconds, resulting in a bit rate of $r_b = 1/T$ bps. Using the Fourier transform, it can be shown that the bandwidth of such a pulse sequence is about $2/T = 2r_b$ Hz.

For example, the digitisation of stereo music at a rate of 44.1 kHz and with each sample quantised to 16 bits generates a bit rate r_b of (2 channels × 44 100 samples/second × 16 bits per sample) 1411.2 kbps. This would require a bandwidth of $2r_b = 2822.4$ kHz. However, using advanced compression and modulation methods the number of bits per second and the required bandwidth can be greatly reduced by a factor of more than 10.

1.4.4 Sampling a Continuous-Time Signal

Figure 1.24 illustrates the process of sampling of a continuous-time signal in time and its effect on the frequency spectrum of the signal. In time-domain a sampled signal can be modelled as the product of a continuous-time signal $x(t)$ multiplied by a periodic impulse train sampler $p(t)$ as

$$\begin{aligned} x_{\text{sampled}}(t) &= x(t)p(t) \\ &= \sum_{m=-\infty}^{\infty} x(t)\,\delta(t - mT_s) \end{aligned} \tag{1.24}$$

where T_s is the sampling interval (the sampling frequency is $F_s = 1/T_s$), $\delta(t)$ is the discrete-time delta (unit-sample) function and the sampling train function $p(t)$ is defined as

$$p(t) = \sum_{m=-\infty}^{\infty} \delta(t - mT_s) \tag{1.25}$$

The spectrum, $P(f)$, of a periodic train of sampling impulses in time $p(t)$, is a periodic train of impulses in frequency given by

$$P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \tag{1.26}$$

where $F_s = 1/T_s$ is the sampling frequency.

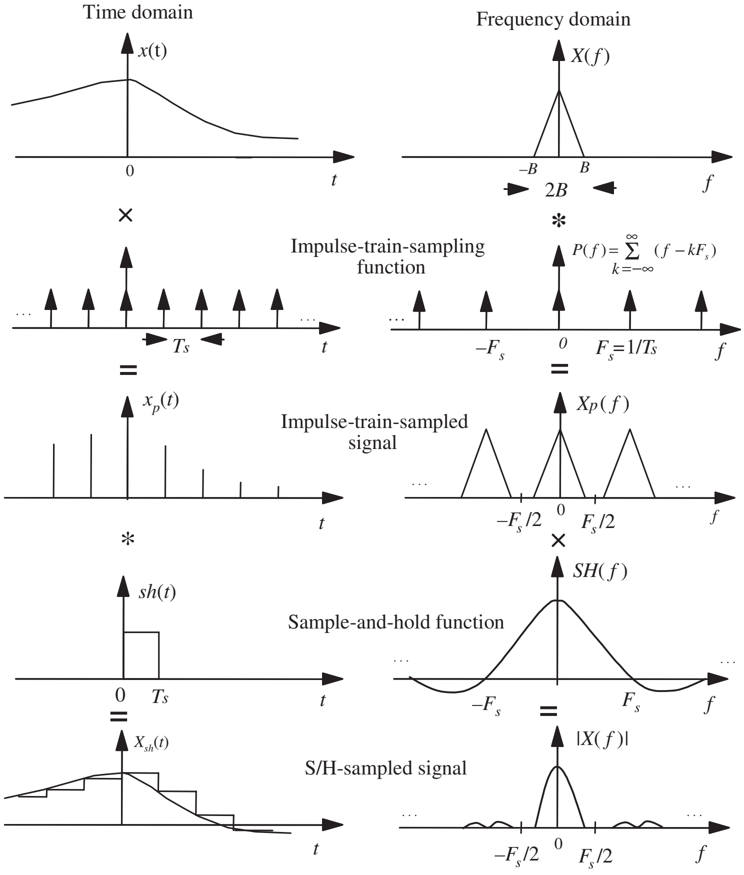


Figure 1.24 A sample-and-hold signal is modelled as an impulse-train sampling followed by convolution with a rectangular pulse.

Since multiplication of two time-domain signals is equivalent to the convolution of their frequency spectra we have

$$X_{\text{sampled}}(f) = FT[x(t).p(t)] = X(f)^{*}P(f) = \sum_{k=-\infty}^{\infty} X(f - kF_s) \tag{1.27}$$

where the operator $FT[.]$ denotes the Fourier transform.

Note from Equation (1.27) that 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, Equation (1.27) shows that the sampling of a signal $x(t)$ results in a periodic repetition of its spectrum $X(f)$ with the 'images' of the baseband spectrum $X(f)$ centred on frequencies $\pm F_s, \pm 2F_s, \ldots$ as shown in Figure 1.24.

Note in Figure 1.24 that a sample-and-hold process produces a sampled signal which is in the shape of an amplitude-modulated staircase function. Also note that the sample-and-hold staircase function can itself be modelled as the output of a filter, with a rectangular impulse response, excited by an idealised sampling impulse train as shown in Figure 1.24.

1.4.5 Aliasing Distortion

The process of sampling results in a periodic repetition of the spectrum of the original signal. When the sampling frequency F_s is higher than twice the maximum frequency content of the signal B Hz (i.e. $F_s > 2B$), then the repetitions ('images') of the signal spectra are separated as shown in Figure 1.24. 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 B Hz. If the sampling frequency is less than $2B$ (i.e. $F_s < 2B$), then the adjacent repetitions of the spectrum overlap and in this case the original spectrum cannot be recovered. The distortion, due to an insufficiently high sampling rate, is irrevocable and is known as *aliasing*. Note in Figure 1.25 that the aliasing distortion results in the high frequency components of the signal folding and appearing at the lower frequencies, hence the name aliasing. Figure 1.26 shows the sum of two sine waves sampled at above and below the Nyquist sampling rate. Note that below the Nyquist rate a frequency of F_0 may appear at $kF_s + F_0$ where k is an integer, as shown in Figure 1.26.

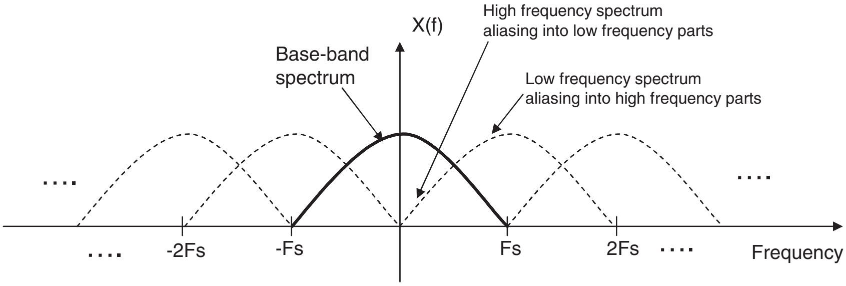

High frequency spectrum aliasing into low frequency parts
X(f)
Base-band spectrum
Low frequency spectrum aliasing into high frequency parts
....
....
....
-2Fs
-Fs
Fs
2Fs
....
Frequency

Figure 1.25 Aliasing distortion results from the overlap of spectral images (dashed curves) with the baseband spectrum. Note high frequency aliases itself as low frequency and vice versa. In this example the signal is sampled at half the required rate.

1.4.6 Nyquist Sampling Theorem

The above observation on aliasing distortion is the basis of the Nyquist sampling theorem, which states: *a band-limited continuous-time signal, with highest frequency content (bandwidth) of B Hz, can be*

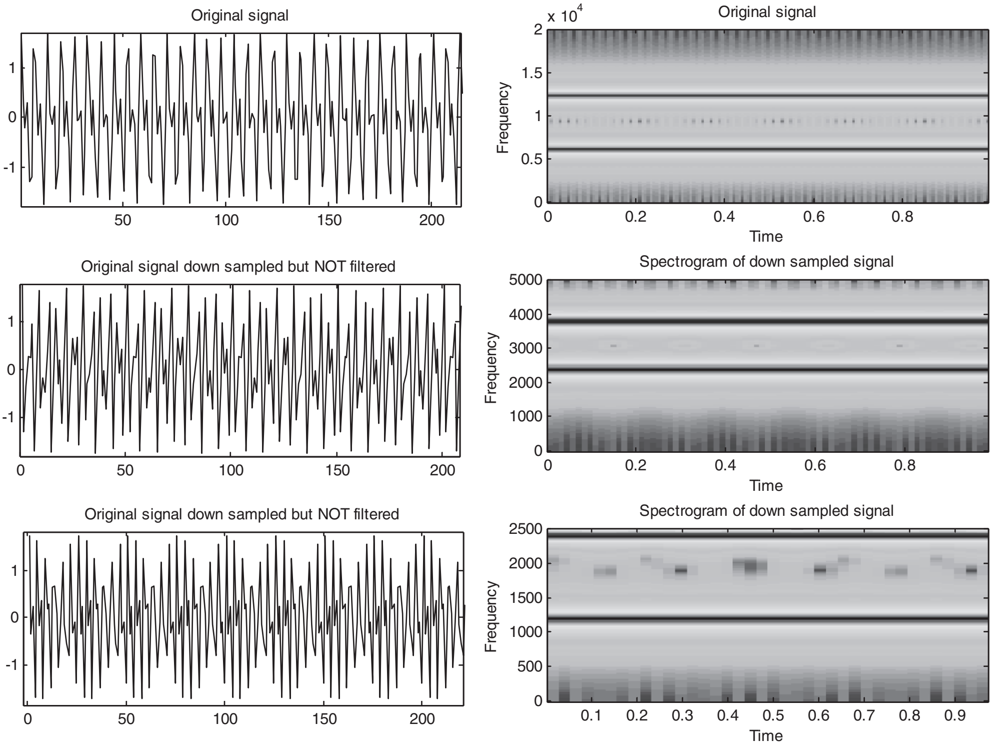


Figure 1.26 Illustration of aliasing. Top panel: the sum of two sinewaves, the assumed frequencies of the sinewaves are 6200 Hz and 12 400 Hz, the sampling frequency is 40 000 Hz. Middle panel: the sine waves down-sampled by a factor of 4 to a sampling frequency 10 000 Hz; note the aliased frequencies appear at $10\,000 - 6200 = 3800$ Hz and $-10\,000 + 12\,400 = 2400$ Hz. Bottom panel: the sine waves down-sampled by a factor of 8 to a sampling frequency of 5000 Hz; note the aliased frequencies appear at $-5000 + 6200 = 1200$ Hz and at $-2 \times 5000 + 12\,400 = 2400$ Hz.

recovered from its samples provided that the sampling frequency F_s is greater than 2B samples per second so that there is no aliasing. Note that the sampling frequency F_s needs to be greater than $2B$ to avoid aliasing distortion and to allow frequency space for the transition band of a low-pass filter which is used to recover the original (baseband) continuous signal from its sampled version.

In practice sampling is achieved using an electronic switch that allows a capacitor to charge or discharge to the level of the input voltage once every T_s seconds as illustrated in Figure 1.23. The sample-and-hold signal can be modelled as the output of a filter with a rectangular impulse response, and with the impulse-train-sampled signal as the input as illustrated in Figure 1.24.

1.4.7 Quantisation

Quantisation is the process of converting each continuous-valued sample of a signal into a discrete value sample that can be assigned a unique digital codeword. For digital signal processing, discrete-time continuous-amplitude samples, from the sample-and-hold, are quantised and mapped into n-bit binary code words before being stored and processing.

Figure 1.27 illustrates an example of the quantisation of a signal into four discrete quantisation levels with each quantisation level represented by a 2-bit codeword. For quantisation to n-bit codewords, the

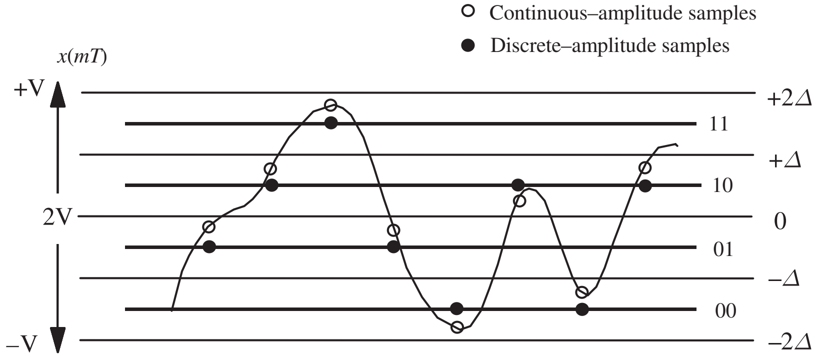


Figure 1.27 Illustration of offset-binary scalar quantisation.

amplitude range of the signal is divided into 2^n quantisation levels. Each continuous-amplitude sample is quantised to the nearest quantisation level and then mapped to the n-bit binary code assigned to that level.

Quantisation is a many-to-one mapping; this means that all the infinite number of values that fall within the continuum of the infinite values of a quantisation band are mapped to one single value at the centre of the band. The mapping is hence an irreversible process in that we cannot recover the exact value of the quantised sample. The mapping between an analogue sample $x_a(m)$ and its quantised value $x(m)$ can be expressed as

$$x(m) = Q\ [x_a(m)] \tag{1.28}$$

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

The performance of a quantiser is measured by signal-to-quantisation noise ratio (SQNR). The quantisation noise is defined as the difference between the analogue value of a sample and its quantised value as

$$e(m) = x(m) - x_a(m) \tag{1.29}$$

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 random process with a uniform probability distribution (i.e. a probability of $1/\Delta$ and with an amplitude range of $\pm\Delta/2$) we can express the noise power as

$$\begin{aligned}\mathcal{E}[e^2(m)] &= \int_{-\Delta/2}^{\Delta/2} p\big(e(m)\big)e^2(m)de(m) = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2(m)de(m) \\ &= \frac{\Delta^2}{12} = \frac{V^2 2^{-2n}}{3}\end{aligned} \tag{1.30}$$

where $\mathcal{E}[\cdot]$ is the expectation or averaging operator and the function $p(e(m)) = 1/\Delta$, shown in Figure (1.28), is the uniform probability density function of the noise and $\Delta = 2V2^{-n}$. Using Equation (1.30) the SQNR is given by

$$\begin{aligned}SQNR(n) &= 10\ \log_{10}\left(\frac{\mathcal{E}\left[x^2(m)\right]}{\mathcal{E}[e^2(m)]}\right) = 10\ \log_{10}\left(\frac{P_{\text{signal}}}{V^2 2^{-2n}/3}\right) \\ &= 10\ \log_{10}\ 3 - 10\ \log_{10}\left(\frac{V^2}{P_{\text{signal}}}\right) + 10\ \log_{10}\ 2^{2n} \\ &= 4.77 - \alpha + 6\ n\end{aligned} \tag{1.31}$$

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}, which for a sine wave α is 3. Therefore, from Equation (1.31) every additional bit in an analogue-to-digital converter results in a 6 dB improvement in signal-to-quantisation noise ratio.

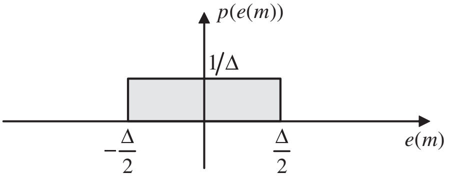


Figure 1.28 Illustration of the uniform probability distribution of the quantization noise.

1.4.8 Non-Linear Quantisation, Companding

A uniform quantiser is only optimal, in the sense of achieving the minimum mean squared error, when the input signal is uniformly distributed within the full range of the quantiser, so that the uniform probability distribution of the signal sample values and the uniform distribution of the quantiser levels are matched and hence different quantisation levels are used with equal probability.

When a signal has a non-uniform probability distribution then a non-uniform quantisation scheme matched to the probability distribution of the signal is more appropriate. This can also be achieved through a transformation of the input signal to change the distribution of the input signal towards a uniform distribution prior to application of a uniform quantiser.

For speech signals, non-uniform qunatisation is achieved through a logarithmic compression of speech, a process known as companding, Figure 1.29. Companding (derived from compressing-expanding) refers to the process of first compressing an analogue signal at the transmitter, and then expanding this signal back to its original size at the receiver. During the companding process, continuous-amplitude input samples are compressed logarithmically and then quantised and coded using a uniform quantiser. The assumption is that speech has an exponential distribution and that the logarithm of speech has a more uniform distribution.

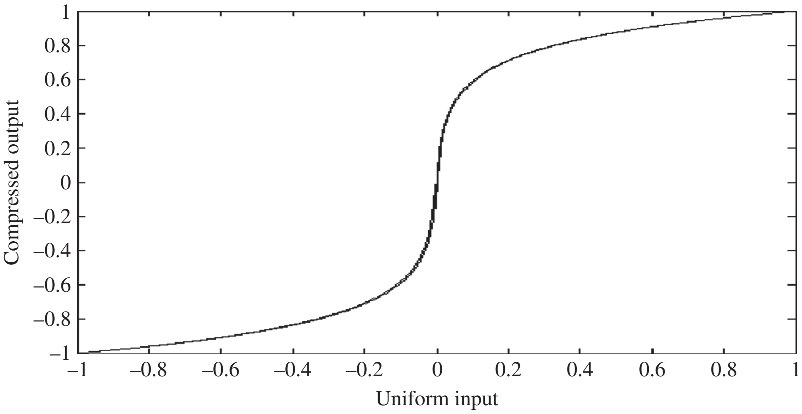


Figure 1.29 Illustration of the compression curves of A-law and u-law quantisers. Note that the curves almost coincide and appear as one.

Figure 1.30 shows the effect of logarithmic compression on the distribution of a Gaussian signal. Note from Figure 1.30(b) that the distribution of the Gaussian signal is more spread after logarithmic compression. Figure 1.31 shows three sets of plots of speech and their respective histogram for speech quantised with 16 bits uniform quantisation, 8 bits uniform quantisation and logarithmic compression

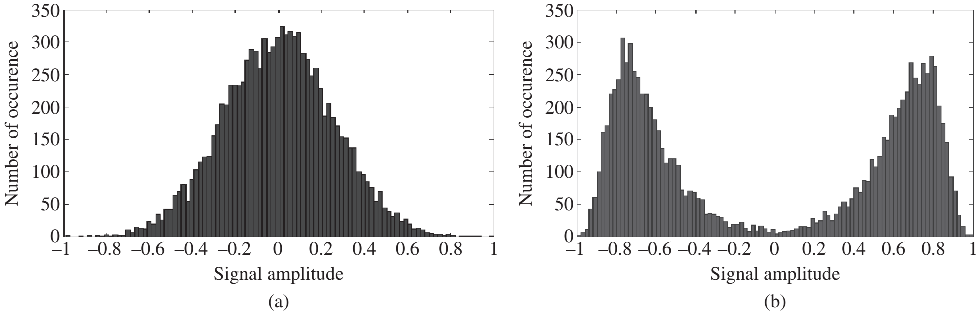


Figure 1.30 (a) The histogram of a Gaussian input signal to a u-law logarithmic function, (b) the histogram of the output of the u-law function.

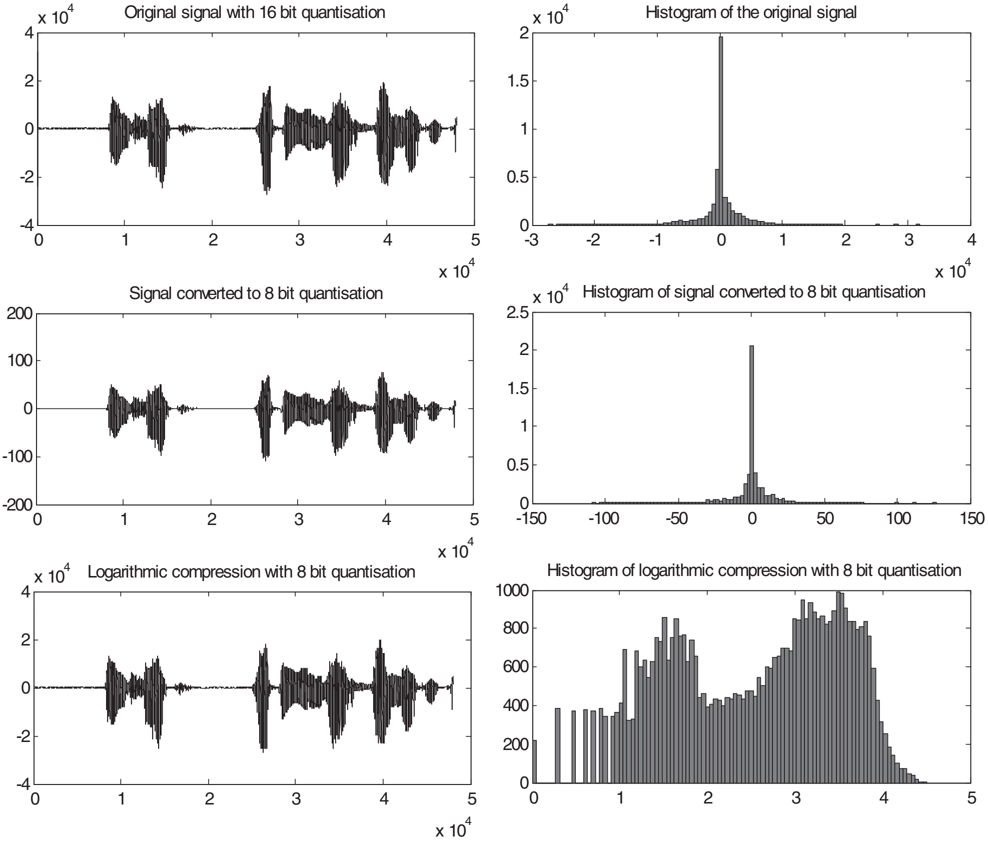


Figure 1.31 From top panel, plots of speech and their histograms quantised with: 16 bits uniform, 8 bits uniform and 8 bits logarithmic respectively.

followed by 8 bits uniform quantisation respectively. Note that the histogram of the logarithm of absolute value of speech has a relatively uniform distribution suitable for uniform quantisation.

The International Telecommunication Union (ITU) standards for companding are called u-law (in the USA) and A-law (in Europe). The u-law compression of a normalised sample value x is given by the relation

$$F(x) = sign(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)} \tag{1.32}$$

where for the parameter μ a value of 255 is typically used. The A-law compression of a sample x is given by the relation

$$F(x) = \begin{cases} \dfrac{A|x|}{1+\ln A} & if \quad 0 \le |x| < \dfrac{1}{A} \\ \dfrac{1+A|x|}{1+\ln A} & if \quad \dfrac{1}{A} \le |x| \le 1 \end{cases} \tag{1.33}$$

A value of 78.6 is used for A. A-law and u-law methods are implemented using 8-bit codewords per sample (256 quantisation levels). At a speech sampling rate of 8 kHz this results in a bit rate of 64 kbps. An implementation of the coding methods may divide a dynamic range into a total of 16 segments: 8 positive and 8 negative segments. The segment range increase logarithmically; each segment is twice the range of the preceding one. Each segment is coded with 4 bits and a further 4-bit uniform quantisation is used within each segment. At 8 bits per sample, A-law and u-law quantisation methods can achieve the equivalent quality of 13-bit uniform quantisation.

1.5 Summary

This chapter began with a definition of signal, noise and information and provided a qualitative explanation of their relationship. A broad categorisation of the various signal processing methodologies was provided. We considered several key applications of digital signal processing in biomedical signal processing, adaptive noise reduction, channel equalisation, pattern classification/recognition, audio signal coding, signal detection, spatial processing for directional reception of signals, Dolby noise reduction, radar and watermarking. Finally this chapter provided an overview of the most basic processes in a digital signal processing system namely sampling and quantisation.

Bibliography

Alexander S.T. (1986) *Adaptive Signal Processing Theory and Applications*. Springer-Verlag, New York.

Cox I. J., Miller M.L., Bloom J.A., Fridrich J. and Kalker T. (2008) *Digital Watermarking and Steganography*, 2nd edn, Morgan Kaufmann.

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*. John Wiley & Sons, Inc, 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. of the ASME, Series D, *Journal of Basic Engineering*, **82**: 35–45.

Kay S.M. (1993) *Fundamentals of Statistical Signal Processing, Estimation* Theory. Prentice-Hall, Englewood Cliffs, NJ.

Kung S.Y. (1993) *Digital Neural Networks*. 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.

Marple S.L. (1987) *Digital Spectral Analysis with Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Nyquist, H., Certain topics in telegraph transmission theory, *AIEE Trans.*, **47**: 617–644, Jan. 1928.

Oppenheim A.V. and Schafer R.W. (1999) *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 Schafer 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.

Shannon C.E. (1948) A Mathematical Theory of Communication. *Bell Systems Tech. J.*, **27**: 379–423, 623–656.

Shannon, C.E. (1949) Communications in the presence of noise, *Proc. IRE*, **37**: 10–21, Jan.

Therrien C.W. (1992) *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ.

Vaidyanathan P.P. (1993) *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

van-trees H.L. (1971) *Detection, Estimation and Modulation Theory*. Parts I, II and III. John Wiley & Sons, Inc, New York.

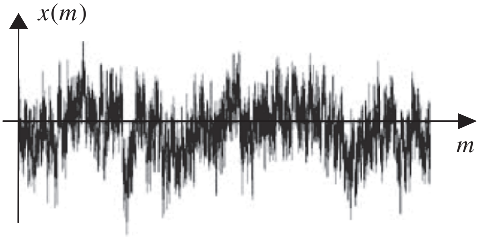
van-trees H.L. (2002) *Detection, Estimation, and Modulation Theory*, Part IV, Optimum Array Processing, John Wiley & Sons, Inc, New York.

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

Noise can be defined as an unwanted signal that interferes with the communication or measurement of another signal. A noise itself is an information-bearing signal that conveys information regarding the sources of the noise and the environment in which it propagates. For example, the noise from a car engine conveys information regarding the state of the engine and how smoothly it is running, cosmic radiation provides information on formation and structure of the universe and background speech conversations in a crowded venue can constitute interference with the hearing of a desired conversation or speech.

The types and sources of noise and distortions are many and varied and include: (i) *electronic noise* such as thermal noise and shot noise, (ii) *acoustic noise* emanating from moving, vibrating or colliding sources such as revolving machines, moving vehicles, keyboard clicks, wind and rain, (iii) *electromagnetic noise* that can interfere with the transmission and reception of voice, image and data over the radio-frequency spectrum, (iv) *electrostatic noise* generated by the presence of a voltage, (v) *communication channel* distortion and fading and (vi) *quantisation noise* and lost data packets due to network congestion.

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 non-ideal characteristics of the communication channel, signal fading reverberations, echo, multipath reflections and missing samples.

Noise and distortion are the main factors that limit the capacity of data transmission in telecommunication and the accuracy of results 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 communication, speech recognition, image processing, medical signal processing, radar, sonar, and in any application where the desired signals cannot be isolated from noise and distortion or observed in isolation. 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, electronic device noise (e.g. thermal noise and shot noise), electromagnetic radio-frequency noise, co-channel radio

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

interference, radio-channel distortion, acoustic and line echoes, multipath reflections, fading, outage 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 telecommunication 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.

2.1.1 Different Classes of Noise Sources and Distortions

Depending on its source and physics, a noise can be described as acoustic, electronic, electromagnetic (radio) and electrostatic. Furthermore in digital communication there are also channel distortions and fading and there may be quanisation noise and lost data due to congested networks or faded signal.

The various forms of noise can be classified into a number of categories, indicating the broad physical nature of the noise and their commonly used categorisation, as follows:

1. *Acoustic disturbances* include:

   1.1 *Acoustic noise* emanates from moving, vibrating, or colliding sources and is the most familiar type of noise present in various degrees in everyday environments. Acoustic noise is generated by such sources as moving vehicles, air-conditioners, computer fans, people talking in the background, wind, rain, etc.

   1.2 *Acoustic feedback and echo*: due to the reflections of sounds from the walls of a room or due to the coupling between microphones and speakers, for example in mobile phones.

2. *Electronic Device Noise – thermal noise, shot noise, flicker noise, burst noise*: these are noise generated in electronic devices and include:

   2.1 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.

   2.2 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.

   2.3 Flicker noise has a spectrum that varies inversely with frequency as $1/f$. It results from a variety of effects in electronic devices, such as impurities in a conductive channel and generation and recombination noise in a transistor due to base current.

   2.4 Burst noise consists of step transitions of as high as several hundred millivolts, at random times and durations.

3. *Electromagnetic noise*: present at all frequencies and in particular at the radio frequency range (kHz to GHz range) where telecommunication systems operate. Electromagnetic noise is composed of a combination of man-made noise sources from electrical devices and natural noise sources due to atmospheric noise and cosmic 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, multipath, echo, and fading*: due to non-ideal 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, multipath effect and fading of signals.

6. *Co-Channel interference*: a form of crosstalk from two different radio transmitters on the same frequency channel. It may be caused by adverse atmospheric conditions causing radio signals to be reflected by the troposphere or by over-crowded radio spectrum.
7. *Missing samples*: segments of a signal may be missing due to variety of factors such as a high burst of noise, signal outage, signal over flow or packet losses in communication systems.
8. *Processing noise*: the noise that results from the digital and analog processing of signals, e.g. quantisation noise in digital coding of speech or image signals, or lost data packets in digital data communication systems.

2.1.2 Different Classes and Spectral/Temporal Shapes of Noise

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*: this is purely random noise that has an impulse autocorrelation function and a flat power spectrum. White noise theoretically contains all frequencies in equal power.

(2) *Band-limited white noise*: this a a noise with a flat power spectrum and a limited bandwidth that usually covers the limited spectrum of the device or the signal of interest. The autocorrelation of this noise is sinc-shaped (a sinc function is $\sin(x)/x$).

(3) *Narrowband noise*: this is a noise process with a narrow bandwidth such as a 50/60 Hz ‘hum’ from the electricity supply.

(4) *Coloured noise*: this is non-white noise or any wideband noise whose spectrum has a non-flat shape; examples are pink noise, brown noise and autoregressive noise.

(5) *Impulsive noise*: this noise consists of short-duration pulses of random amplitude, time of occurrence and duration.

(6) *Transient noise pulses*: these consist of relatively long duration noise pulses such as clicks, burst noise etc.

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

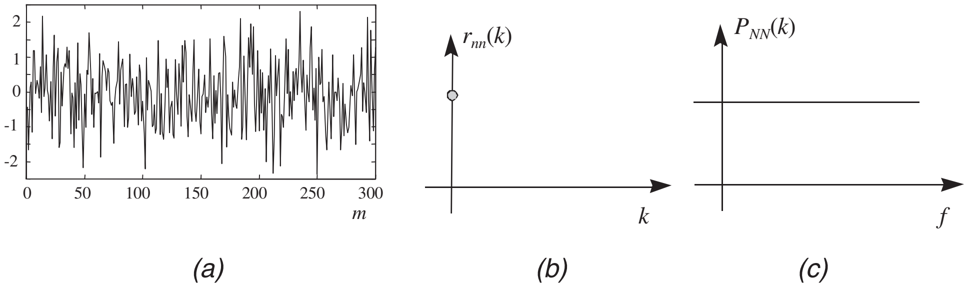


Figure 2.1 Illustration of (a) white noise time-domain signal, (b) its autocorrelation function is a delta function, and (c) its power spectrum is a constant function of frequency.

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 a white noise.

The autocorrelation function of a continuous-time zero-mean white noise process, $n(t)$, with a variance of σ_n^2 is a delta function (Figure 2.1(b)) given by

$$r_{nn}(\tau) = \mathcal{E}[n(t)n(t+\tau)] = \sigma_n^2 \delta(\tau) \tag{2.1}$$

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

$$P_{NN}(f) = \int_{-\infty}^{\infty} r_{nn}(t) e^{-j2\pi ft}\, dt = \sigma_n^2 \tag{2.2}$$

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

A 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

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

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

$$r_{nn}(T_s k) = 2B\sigma_n^2 \frac{\sin(2\pi BT_s k)}{2\pi BT_s k} \tag{2.4}$$

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

$$r_{NN}(T_s k) = 2B\sigma_n^2 \frac{\sin(\pi k)}{\pi k} = 2B\sigma_n^2 \delta(k) \tag{2.5}$$

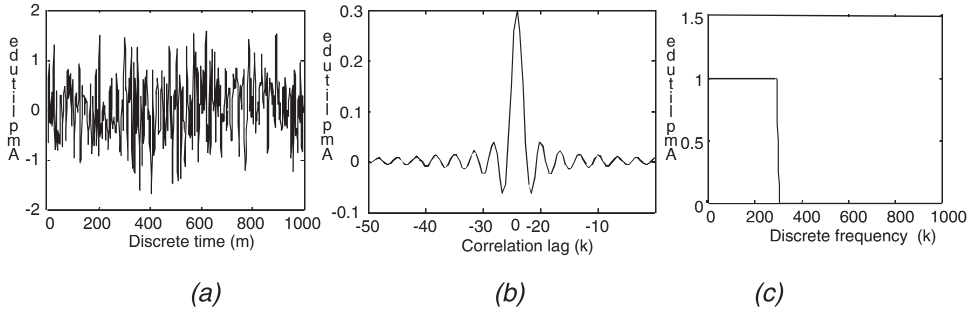


Figure 2.2 Illustration of (a) an oversampled bandlimited white noise signal, (b) its autocorrelation function is a sinc function, and (c) the spectrum of oversampled signal.

In Equation (2.5) the autocorrelation function is a delta function. Figure 2.2 shows a bandlimited signal that has been (over) sampled at more than three times Nyquist rate together with the auto-correlation and power spectrum of the signal.

2.3 Coloured Noise; Pink Noise and Brown Noise

Although the concept of white noise provides a reasonably realistic and mathematically convenient and useful approximation to some predominant noise processes encountered in telecommunication systems, many other noise processes are non-white.

The term coloured noise refers to any broadband noise with a non-white 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, have a non-white predominantly low-frequency spectrum. Furthermore, 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 the so-called pink noise and brown noise, shown in Figures 2.3 and 2.4.

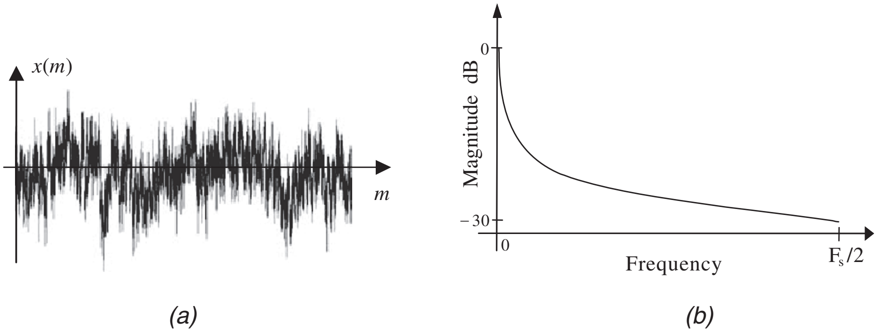


Figure 2.3 (a) A pink noise signal and (b) its magnitude spectrum.

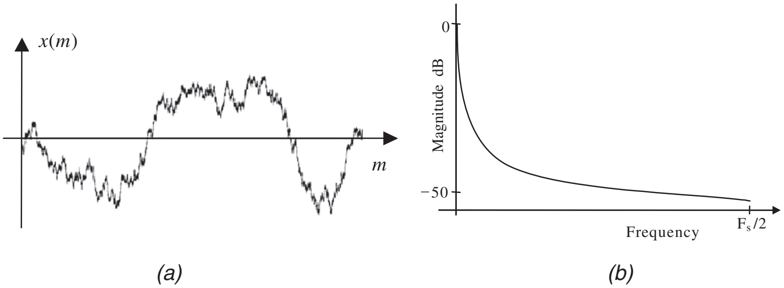


Figure 2.4 (a) A brown noise signal and (b) its magnitude spectrum.

2.4 Impulsive and Click 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 environments in a communication system, drop-outs or surface degradation of audio recordings, clicks from computer keyboards, etc.

Figure 2.5(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 milliseconds (60 samples at a 20 kHz sampling rate) may be considered as impulsive noise. Figures 2.5(b) and (c) illustrate two examples of short-duration pulses and their respective spectra.

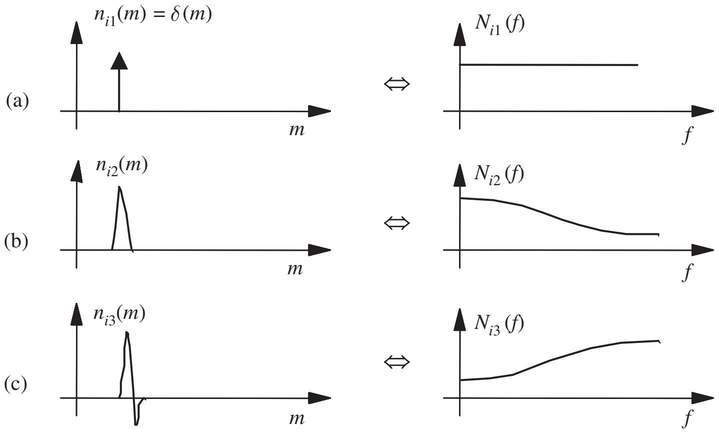


Figure 2.5 Time and frequency sketches of: (a) an ideal impulse, (b) and (c) short-duration pulses.

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 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 non-linear, stationary or time varying. Furthermore, many communication systems exhibit a non-linear characteristic in response to a large-amplitude impulse.

Figure 2.6 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 the channel characteristics with the amplitude of impulsive noise. For example, in Figure 2.6(c) a large impulse excitation has generated a decaying transient pulse with time-varying period. These variations may be attributed to the non-linear characteristics of the playback mechanism.

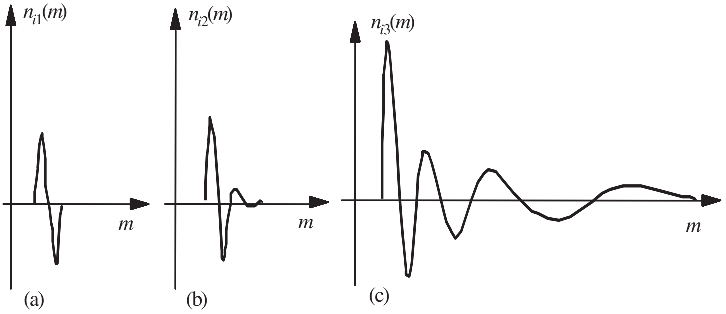


Figure 2.6 Illustration of variations of the impulse response of a non-linear system with the increasing amplitude of the impulse.

2.5 Transient Noise Pulses

Transient noise pulses, observed in most communication systems, are bursts of noise, or long clicks caused by interference or damage to signals during storage or transmission. Transient noise pulses often consist of a relatively short sharp initial pulse followed by decaying low-frequency oscillations as shown in Figure 2.7. 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.

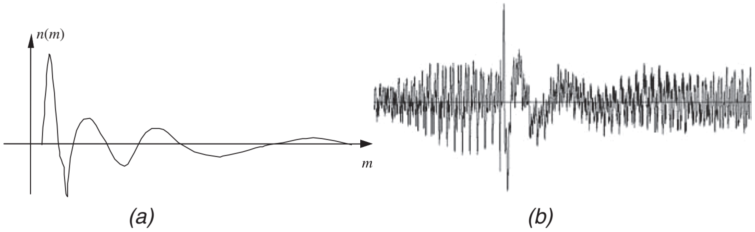


Figure 2.7 (a) A scratch pulse and music from a gramophone record. (b) The averaged profile of a gramophone record scratch pulse.

In a telecommunication 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 characterise the transient noise pulses with a similar degree of consistency as in characterising 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 Figures 2.7(a) and (b). Scratch noise pulses are acoustic manifestations of the response of the stylus and the associated electro-mechanical 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 2.7(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 2.7(b) that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the non-linear modes of response of the electro-mechanical 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 some detail in Chapter 13.

2.6 Thermal Noise

Thermal noise, also referred to as Johnson noise (after its discoverer J.B. Johnson), is a type of electronic noise generated by the random movements of thermally energised (agitated) particles inside an electric conductor. Thermal noise has a white (flat) spectrum. It 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 additional random fluctuations known as shot noise and flicker noise, 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 absent of a voltage electrons move in all different directions. As the temperature of a conductor, due to 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

$$\overline{v^2} = 4kTRB \tag{2.6}$$

where $k = 1.38 \times 10^{-23}$ joules per degree Kelvin 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

$$P_N = \overline{i^2}R = \left(\frac{v_{\text{rms}}}{2R}\right)^2 R = \frac{\overline{v^2}}{4R} = kTB \qquad (\text{W}) \tag{2.7}$$

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

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

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 is a type of electronic noise that arises from the fact that an electronic or photonic current is composed of a random number of discrete electrons or photons with random time of arrivals. Shot noise has a white spectrum and can be modelled by a Poisson probability model, as described in Section 3.7.5.

Note that a given value of electronic current effectively specifies the average value of flow of particles and the actual number of flow of charged particles is a random variable that varies about the average value.

The strength of shot noise increases with the increasing average current flowing through the conductor. However, since the magnitude of the average signal increases more rapidly than that of the shot noise, shot noise is often only a problem with small electronic currents or photonic light intensities.

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 arrival 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) due to temperature and happens without any applied voltage and any average current flowing. Whereas shot noise 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 constitute 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.

For a mathematical expression relating shot noise to average current value I, consider an electric current as the flow of discrete electric charges (electrons). As explained the flow of electrons is not smooth and there will be random fluctuations in the form of shot noise. If the electronic charges act independently of each other, it can be shown that noise current is given by

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

where $e = 1.6 \times 10^{-19}$ coulomb is the electron charge, I is the current and B is the measurement bandwidth. For example, a ‘steady’ current I of 1 amp in a bandwidth 1 MHz has an rms fluctuation of 0.57 microamps. From Equation (2.9) note that shot noise is proportional to the square root of the current, hence the ratio of the signal power to shot noise power has the same value as the direct current value I. Equation (2.9) assumes that the charge carriers making up the current act independently. That is the case for charges crossing a barrier, as for example the current in a junction diode, where the charges move by diffusion; but it is not true for metallic conductors, where there are long-range correlations between charge carriers.

2.8 Flicker (I/*f*) Noise

Flicker noise is an electronic device noise that occurs due to the random fluctuations of electron flow that accompanies direct current in electronic devices such as transistors and vacuum tubes. It results from a variety of effects, such as crystal surface defects in semiconductors, impurities in a conductive channel, generation and recombination noise in a transistor due to base current etc. Flicker noise is more prominent in field effect transistors (FETs) and bulky carbon resistors.

In contrast to the white spectrum of thermal noise and shot noise, flicker noise has a pink-shaped spectrum that varies with $N(f) = 1/f$ as shown in Figure 2.8. The power spectrum of flicker noise resulting from a direct current flow may be modelled (Van Der Ziel, 1970) as

$$N_{\frac{1}{f}}(f) = K_f \frac{I^A f}{f} \tag{2.10}$$

Where I is the value of the current in the electronic device and K_f and A_f are parameters that can be estimated from observations of the noise. A characteristic parameter of the flicker noise is the corner frequency F_c defined as the frequency at which the magnitude spectrum of flicker noise is equal to, and crosses, that of white noise; for frequencies above F_c flicker noise goes below the white noise.

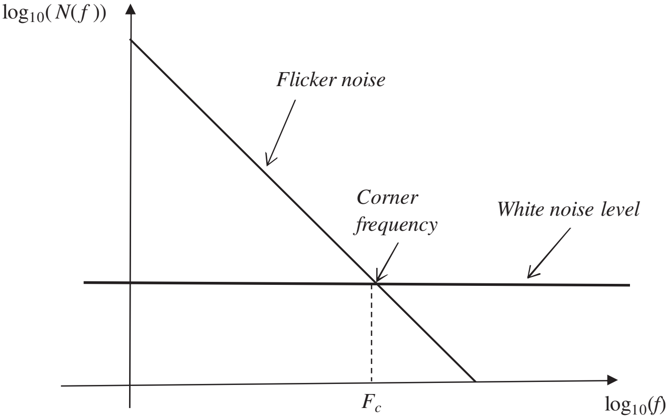


Figure 2.8 (a) A sketch of the shape of spectrum of flicker noise.

In electronic devices, flicker noise is usually a low-frequency phenomenon; as the higher frequencies are drown by white noise from other sources. However, in oscillators, the low-frequency noise can be modulated and shifted to higher frequencies by the oscillator carrier.

Since flicker noise is related to direct current, it may be kept low if the level of direct current is kept low, for example in resistors at low current levels thermal noise will be the predominant effect.

2.9 Burst Noise

Burst noise (also known as popcorn noise) is a type of electronic noise that occurs in semiconductors devices. It consists of pulse-like transitions of voltage to levels as high as several hundred microvolts, at random and unpredictable times and durations.

Each burst shift in offset voltage or current lasts for several milliseconds, and the intervals between the bursts is usually in the low frequency audio range (less than 100 Hz), for this reason burst noise is also known as the popcorn noise due to the popping or crackling sounds it produces in audio circuits.

Burst noise may also be caused by electromagnetic interference such as the interference due to a lightning storm or from the switching of a fluorescent light, microwave, TV or other electro-magnetic systems.

Figure 2.9 shows a sketch of a sequence of burst noise. Burst noise is defined by the following statistical parameters; the duration, the frequency of occurrence, the time of occurrence and the amplitude of bursts.

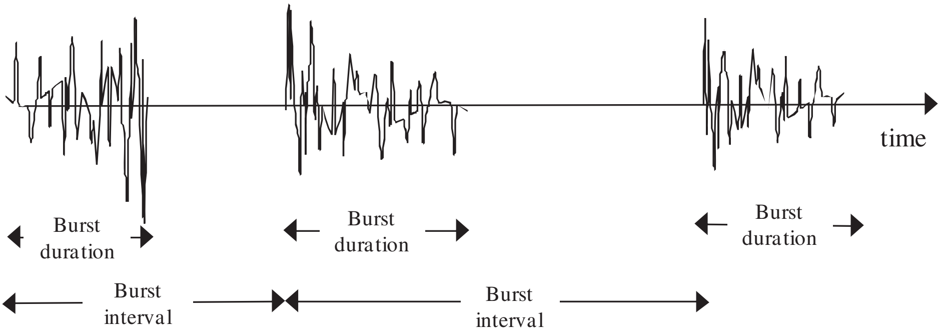


Figure 2.9 (a) A sketch of occurrences of burst noise.

In some situations a burst of noise may itself be composed of a series of impulsive noises in which case the number of impulses within the burst, their intervals, time of occurrence and amplitudes also need to be statistically modelled.

Burst noise can be modelled by a 2-state hidden Markov model (HMM) described in Chapter 5. In some models the number of occurrence of burst noise is modelled by a Poisson probability density whereas the amplitude may be modelled by a Gaussian mixture model.

2.10 Electromagnetic (Radio) Noise

Electromagnetic waves present in the environment constitute a level of background noise that can interfere with the operation of radio communication systems. Electromagnetic waves may emanate from man-made devices or natural sources.

In the order of increasing frequency (or decreasing wavelength) various types of electromagnetic radiation include: electric motors (kHz) radio waves (kHz-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 gamma radiation (10^{23} Hz).

2.10.1 Natural Sources of Radiation of Electromagnetic Noise

The natural source of radiation of electromagnetic waves include atmospheric discharges from thunderstorm lightening, solar and cosmic radio noise radiations and varying electro-magnetic fields of celestial bodies in the magnetosphere.

The range of frequency of non-ionizing natural EM noise includes a huge bandwidth of 10^{14} Hz from below 10^{-3} Hz at the low end of the spectrum to 3×10^{11} Hz at the high end. At the lowest end in the range of .001–3 Hz radio noise are generated in the magnetosphere space cavity. Note that a .003 Hz EM wave has a wavelength equal to (speed of light divided by frequency) $c/f = 3 \times 10^8/.003 = 10^{11}$ meters, which is in the order of galactic distances.

Within the earth's atmosphere the EM wave is generated by an estimated 16 million lightning storms a year. On average each lightning generates 10s of kilo amps of currents and 100s of megawatts of electrical powers. The EM waves generated by lightning have a bandwidth from extremely low frequency to very high frequency of more than 500 MHz. The very low frequencies parts of the lightning's EM waves are trapped by the earth-ionosphere cavity which acts as a low-loss waveguide with resonances at 7.8 Hz and its harmonics of 15.6 Hz, 23.4 Hz and 31.2 Hz. This effect is known as the Schumann effect. At very low frequencies the lightning also produce a series of noise known as spherics (or sferics) there are low frequency impulsive noise, chrips and other signals known by such names clicks, pops, whistlers and chorus. At high radio frequency the average effect of atmospheric EM discharges is white background EM noise.

Atmospheric noise is the main source of natural radiation noise up to high frequency (HF) range of 3 MHz. In HF (3–30 MHz) and very high frequency (VHF) range of 30–300 MHz both atmospheric and cosmic radiations are the sources of noise. Above VHF in the range 300 MHz to 300 GHz the main source of noise is cosmic radiations.

Above 300 GHz, electromagnetic radiation is absorbed by the Earth's atmosphere to such an extent that the atmosphere is opaque to the higher frequencies of electromagnetic radiation. The atmosphere becomes transparent again in the infrared and optical frequency ranges.

2.10.2 Man-made Sources of Radiation of Electromagnetic Noise

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, ac power lines, motors and motor starters, generators, relays, oscillators, fluorescent lamps, and medical devices.

Electrical noise from these sources can be categorised into two basic types: electrostatic and magnetic. These two types of noise are fundamentally different, and thus require different noise-shielding measures. However, 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 communication systems are the radiations from the antennae 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 antennae, as described in Chapter 17.

2.11 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 multi-path 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 equalisation are essential components of modern digital communication systems. Channel equalisation is particularly important in modern cellular communication systems, since the variations of channel characteristics and propagation attenuation in cellular radio systems are far greater than those of the landline systems.

Figure 2.10 illustrates the frequency response of a channel with one invertible and two non-invertible regions. In the non-invertible regions, the signal frequencies are heavily attenuated and lost to the channel

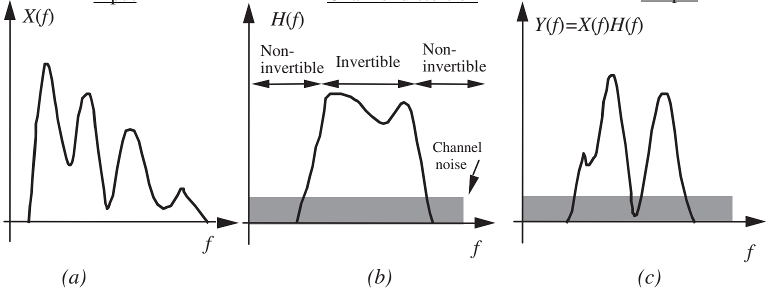


Figure 2.10 Illustration of channel distortion: (a) the input signal spectrum, (b) the channel frequency response, (c) the channel output.

noise. In the invertible region, the signal is distorted but recoverable. 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 SNR. Channel equalization is covered in detail in Chapter 15.

2.12 Echo and Multi-path 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 changes. Multipath and echo happen for both acoustic and electromagnetic signals.

Echo implies that a part of the signal is returned back 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 4-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 communication 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 communication environments multipath propagation can cause severe distortion of the signals if it is not modelled and compensated. Chapter 17 provides an introduction to multipath effects in mobile communication systems.

2.13 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.

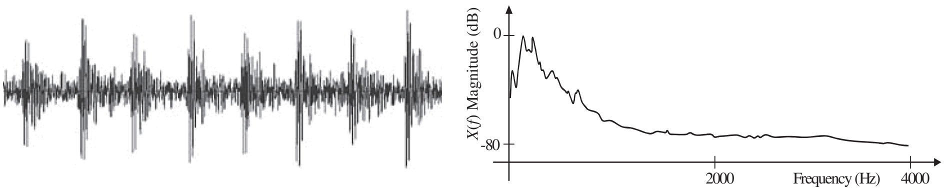
The simplest method for noise modelling, often used in current practice, is to estimate the noise statistics from the signal-inactive periods. In optimal Bayesian signal processing methods, a set of probability models, such as hidden Markov models (HMMs), or Gaussian mixture models (GMMs) are trained 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 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.

2.13.1 Frequency Analysis and Characterisation of 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 2.11 illustrates the noise from an electric drill, which, as expected, has a periodic structure. The spectrum of the drilling noise shown in Figure 2.11(b) 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 2.12(a) and (b) show examples of the spectra of car noise recorded from a BMW and a Volvo respectively. The noise in a car is non-stationary, and varied, and may include the following sources: quasi-periodic noise from the car engine and the revolving mechanical parts of the car:

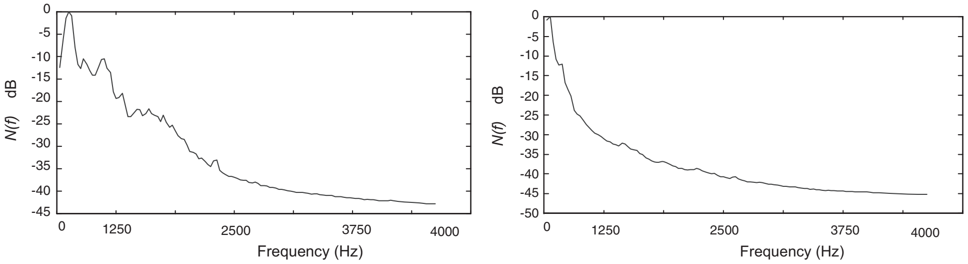
(1) noise from the surface contact of wheels and the road surface;
(2) noise from the air flow into the car through the air ducts, windows, sunroof, etc;
(3) noise from passing/overtaking vehicles.

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



(a) *(b)*

Figure 2.11 Illustration of: (a) the time-waveform of a drill noise, and (b) the frequency spectrum of the drill noise.



(a) *(b)*

Figure 2.12 Power spectra of car noise: (a) a BMW at 70 mph and (b) a Volvo at 70 mph.

Figure 2.13 illustrates the variations of the envelope 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 envelope.

2.13.2 Additive White Gaussian Noise Model (AWGN)

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 non-stationary and non-Gaussian and hence cannot be modelled using the AWGN assumption. Non-stationary and non-Gaussian noise processes can be modelled by a Markovian chain of stationary sub-processes as described briefly in the next section and in detail in Chapter 5.

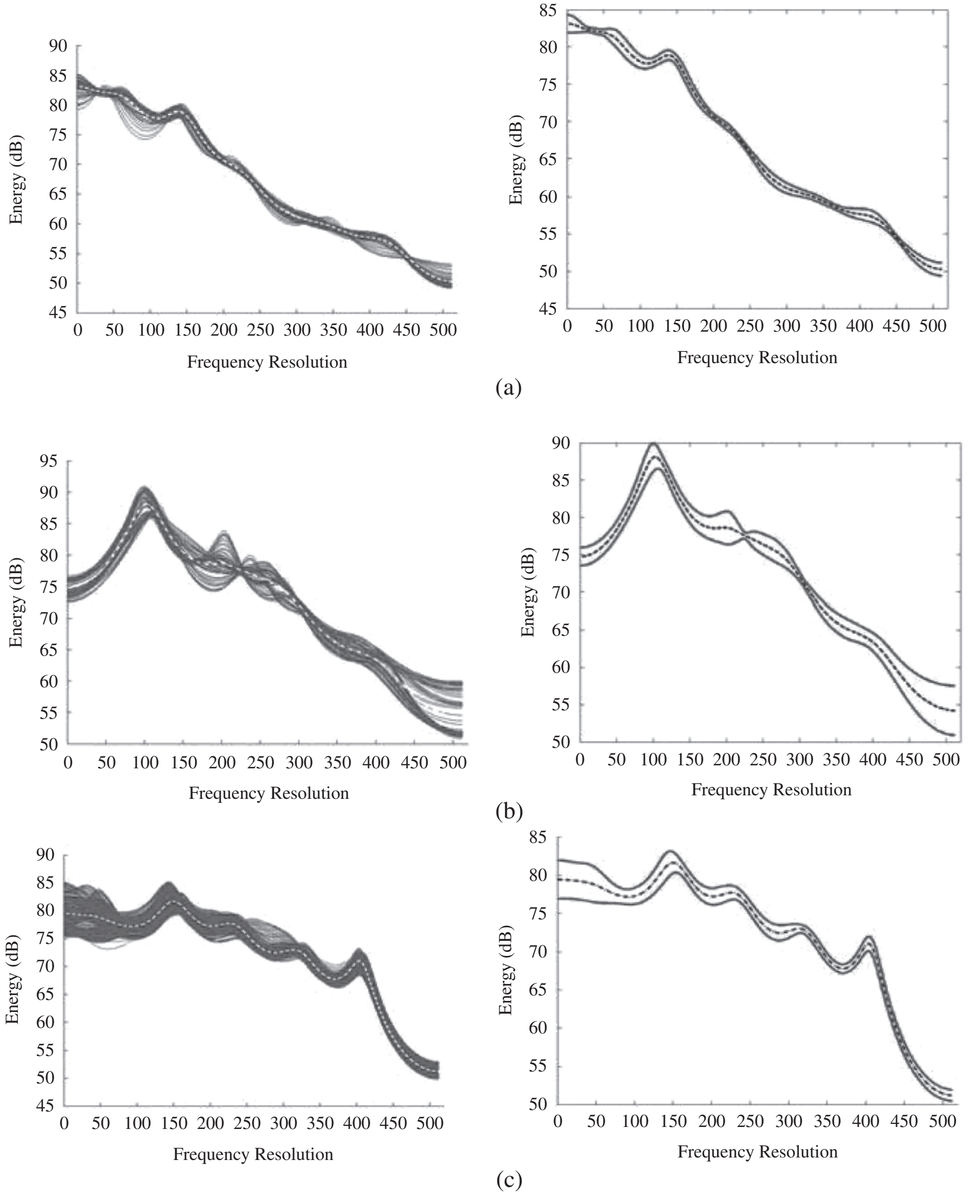


Figure 2.13 Illustration of the mean (left) and standard deviation (right) of the magnitude spectra of: (a) car noise, (b) train noise and (c) street noise.

2.13.3 Hidden Markov Model and Gaussian Mixture Models for Noise

Processes – such as impulsive noise, burst noise, car noise etc – are non-stationary; that is the statistical parameters of the noise, such as its mean, variance and power spectrum, vary with time. Non-stationary processes may be modelled using the hidden Markov models (HMMs) described in detail in Chapter 5.

An HMM is essentially a finite-state Markov chain of stationary sub-processes. The implicit assumption in using HMMs for noise is that the noise statistics can be modelled by a Markovian chain of stationary sub-processes. Note that a stationary noise process can be modelled by a single-state HMM. For a non-stationary noise, a multi-state 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 non-stationary character of the noise.

An example of a non-stationary noise is the impulsive noise of Figure 2.14(a). Figure 2.14(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 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.

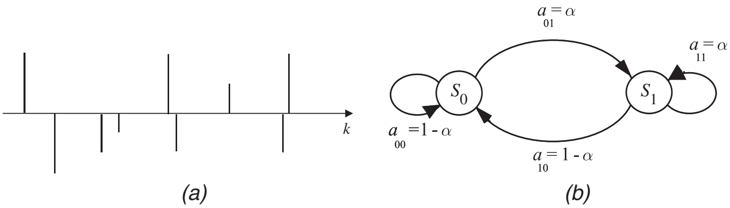


Figure 2.14 (a) An impulsive noise sequence. (b) A binary-state model of impulsive noise.

When the noise signal does not have a well-defined Markovian structure, a Gaussian mixture model, described in Section 4.5, can be used. Gaussian mixture models can also be thought of a single state hidden Markov model.

Bibliography

Bell D.A. (1960) *Electrical Noise: Fundamentals and Physical Mechanism*. Van Nostrand, London.

Bennett W.R. (1960) *Electrical Noise*. McGraw-Hill. NewYork.

Blackard K.L., Rappaport T.S. and Bostian C.W. (1993) Measurements and Models of Radio Frequency Impulsive Noise for Indoor Wireless Communications, *IEEE J. Select. Areas Commun.*, **11**: 991–1001.

Chandra A. (2002) Measurements of Radio Impulsive Noise from various Sources in an Indoor Environment at 900 MHz and 1 800 MHz, *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, **2**: 639–643.

Campbell N., (1909) The study of discontinuous phenomena, *Proc Cambridge. 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.

Garcia Sanchez M., de Haro L., Calvo Ramón M., Mansilla A., Montero Ortega C. and Oliver D. (1999) 'Impulsive Noise Measurements and Characterization in a UHF Digital TV Channel'; *IEEE Transactions on electromagnetic compatibility*, **41**(2): 124–136.

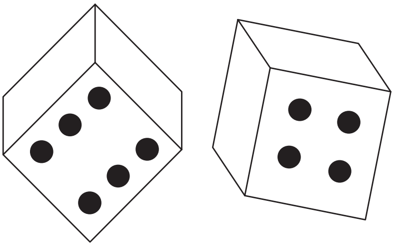
Godsill S.J. (1993) The Restoration of Degraded Audio Signals. Ph.D. Thesis, Cambridge University.

Rice, S.O. (1944) Mathematical analysis of random noise, *Bell Syst. Tech J*, **23**: 282–332, [Reprinted in *Selected Papers on Noise and Stochastic Processes*, N. Wax, Ed. New York: Dover, 1954, pp. 133–294].

Schwartz M. (1990) *Information Transmission, Modulation and Noise*. 4th edn, McGraw-Hill, New York.

Van Der Ziel A. *Noise, Sources, Characterization, Measurement*, Prentice Hall, 1970.

Van-Trees H.L. (1971) *Detection, Estimation and Modulation Theory*. Parts I, II and III. John Wiley & Sons, Inc, New York.



3

Information Theory and Probability Models

Information theory and probability models provide the mathematical foundation for the analysis, modelling and design of telecommunication and signal processing systems.

What constitutes information, news, data or knowledge may be somewhat of a philosophical question. However, information theory is concerned with quantifiable variables, hence, information may be defined as knowledge or data about the states or values of a random process, such as the number of states of the process, the likelihood of each state, the probability of the observable outputs in each state, the conditional probabilities of the state sequences (i.e. how the random process moves across its various states) and the process history (i.e. its past, current and likely future states).

For example, the history of fluctuations of random variables, such as the various states of weather/climate, the demand on a cellular mobile phone system at a given time of day and place or the fluctuations of stock market prices, may be used to obtain a finite-state model of these random variables.

Information theory allows for the prediction and estimation of the values or states of a random process, from related observations that may be incomplete and/or noisy. This is facilitated through the utilisation of the probability models of the information-bearing process and noise and the history of the dependencies of the state sequence of the random 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 diverse areas of applications such as speech/image recognition, audio/video coding, bio-engineering, 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 processes 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 to Huffman coding is presented. Finally, several different forms of probability models and their applications in communication signal processing are considered.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi

3.1 Introduction: Probability and Information Models

All areas of information processing and decision making deal with signals that are random, which may carry multiple layers of information (e.g. speech signals convey words, meaning, gender, emotion, state of health, accent etc.) and which are often noisy and perhaps incomplete.

Figure 3.1 illustrates a simplified bottom-up illustration of the information processing hierarchy from the signal generation level to information decoding and decision making. At all levels of information flow there is some randomness or uncertainty and observation may contain mixed signals and/or hidden parameters and noise.

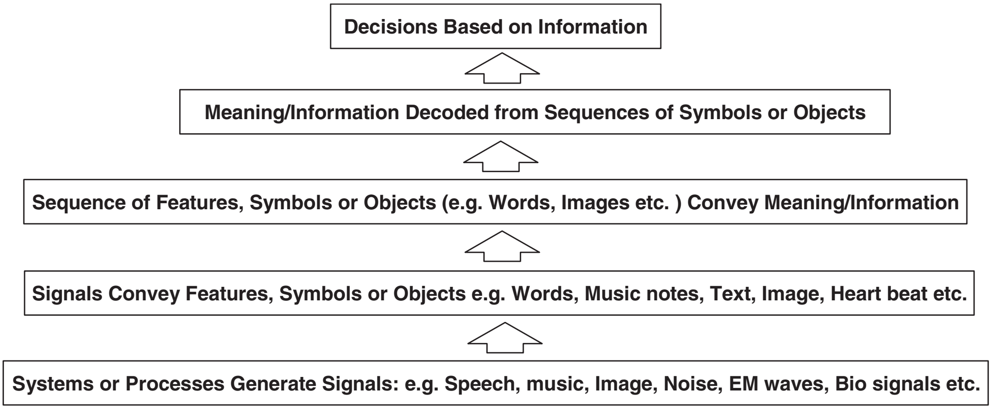

Decisions Based on Information
Meaning/Information Decoded from Sequences of Symbols or Objects
Sequence of Features, Symbols or Objects (e.g. Words, Images etc.) Convey Meaning/Information
Signals Convey Features, Symbols or Objects e.g. Words, Music notes, Text, Image, Heart beat etc.
Systems or Processes Generate Signals: e.g. Speech, music, Image, Noise, EM waves, Bio signals etc.

Figure 3.1 A bottom-up illustration of the information processing hierarchy from the signal generation to information extraction and decision making. At all levels of signal/information processing there is some randomness or uncertainty or noise that may be modelled with probability functions.

Probability models form the foundation of information theory and decision making. As shown in Figure 3.2, many applications of information theory such as data compression, pattern recognition, decision making, search engines and artificial intelligence are based on the use of probability models of the signals.

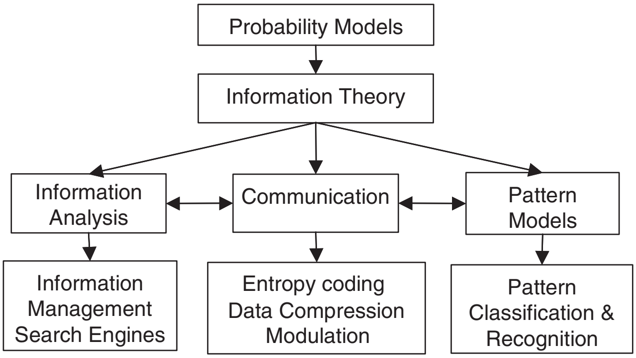

Probability Models
Information Theory
Information Analysis
Communication
Pattern Models
Information Management Search Engines
Entropy coding Data Compression Modulation
Pattern Classification & Recognition

Figure 3.2 A simplified tree-structure illustration of probability models leading to information theory and its applications in information analysis, communication and pattern recognition and specific examples of the applications.

As explained later in this chapter the information content of a source is measured and quantified in units of ‘bits’ in terms of a logarithmic function of probability known as entropy. It would be practically impossible to design and develop large-scale efficient reliable advanced communication systems without the use of probability models and information theory.

Information theory deals with information-bearing signals that are random such as text, speech, image, noise and stock market 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 entirely predictable from the past values.

The modelling, quantification, estimation and ranking of information in communication systems and search engines requires appropriate mathematical tools for modelling the randomness and uncertainty in information-bearing signals and the main mathematical tools for modelling randomness/uncertainty in a signal are those offered by statistics and probability theory.

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, bio-medical engineering, 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 partially characterised with relatively simple statistics, such as the mean, the correlation and the power spectrum. We study stationary, non-stationary 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 Processes

This section introduces the concepts of random and stochastic processes and describes a method for generation of random numbers.

3.2.1 Information-bearing Random Signals vs Deterministic Signals

Signals, in terms of one of their most fundamental characteristics, i.e. their ability to convey information, can be classified into two broad categories:

(1) *Deterministic* signals such as sine waves that on their own convey no information but can act as carriers of information when modulated by a random information-bearing signal.
(2) *Random* signals such as speech and image that contain 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 multi-dimensional.

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 characteristic parameters.

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 in part 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 telecommunication it is wasteful 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, correlation, 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.

Example 3.1 A deterministic signal model

Figure 3.3(a) shows a model of a deterministic discrete-time signal. The model generates an output signal $x(m)$ from the P past output samples as

$$x(m) = h_1\big(x(m-1),\ x(m-2), \ldots,\ x(m-P)\big) + \delta(m) \tag{3.1}$$

where the function h_1 may be a linear or a non-linear 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 are all that is required to generate, or predict the future values of the signal $x(m)$.

For example, for a discrete-time sinusoidal signal generator (i.e. a *digital oscillator*) Equation (3.1) becomes

$$x(m) = a_1 x(m-1) + a_2\, x(m-2) + \delta(m) \tag{3.2}$$

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.

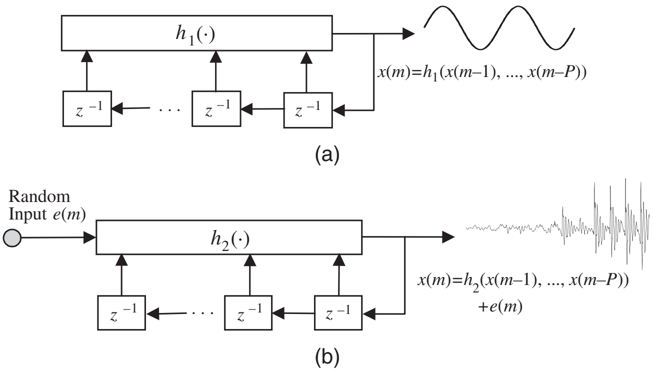


Figure 3.3 (a) A deterministic signal model, (b) a random signal model.

Example 3.2 A random signal model

Figure 3.3(b) illustrates a model for a random signal given by

$$x(m) = h_2\big(x(m-1),\ x(m-2), \ldots, x(m-P)\big) + e(m) \tag{3.3}$$

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

$$x(m) = a_1 x(m-1) + a_2 x(m-2) + e(m) \tag{3.4}$$

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

3.2.2 Pseudo-Random Number Generators (PRNG)

Random numbers are generated by a feedback system such as $x(m) = f(x(m-1), x(m-2), \ldots)$ as shown in Figure 3.4. The random number generation starts with an initial value, $x(-1)$, known as the 'seed'.

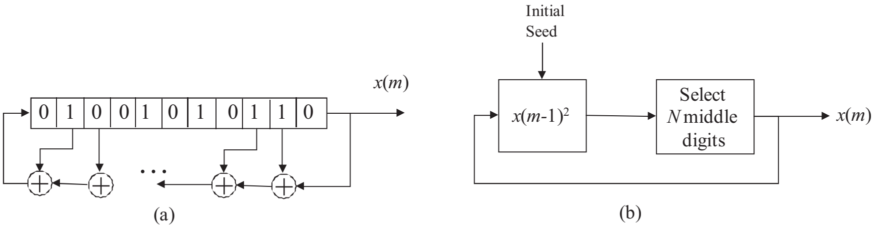


Figure 3.4 Illustration of two different PRNG methods: (a) a linear feedback shift register method and (b) a simple middle digits of power of 2 method.

A 'random' number generator implemented on a deterministic computer system, with finite memory, will exhibit periodic properties and hence it will not be purely random. This is because a computer is a finite-state memory system and will, given sufficient time, eventually revisit previous internal states, after which it will repeat the state sequence. For this reason computer random number generators are known as pseudo-random number generators (PRNG).

The outputs of pseudo-random number generators only approximate some of the statistical properties of random numbers. However, in practice the undesirable periodicity can be 'avoided' if the period is extremely large so that no repetitions would be practically observed. Note that the length of the maximum period typically doubles with each additional bit, and hence it is not difficult to build PRNGs with periods so long that the repetitions will not observed.

A PRNG can be started from an arbitrary starting point, using a 'random' seed state; it will then always produce the same sequence when initialised with that state.

Most pseudo-random generator algorithms produce sequences which are uniformly distributed. Common classes of PRNG algorithms are linear feedback shift registers (Figure 3.4(a)), linear congruential generators and lagged Fibonacci generators. Recent developments of PRNG algorithms include Blum Blum Shub, Fortuna, and the Mersenne twister. A description of these methods is outside the scope of this book.

The basic principles of PRNG can be illustrated using a simple procedure, illustrated in Figure (3.4.b), that starts from an initial N-digit seed and squares the number to yield a $2N$ digit output, the N middle digits of the output are fed back to the square function. At each cycle the program outputs M digits where $M \leq N$. Figure 3.5 shows the output of the PRNG together with its histogram and autocorrelation function.

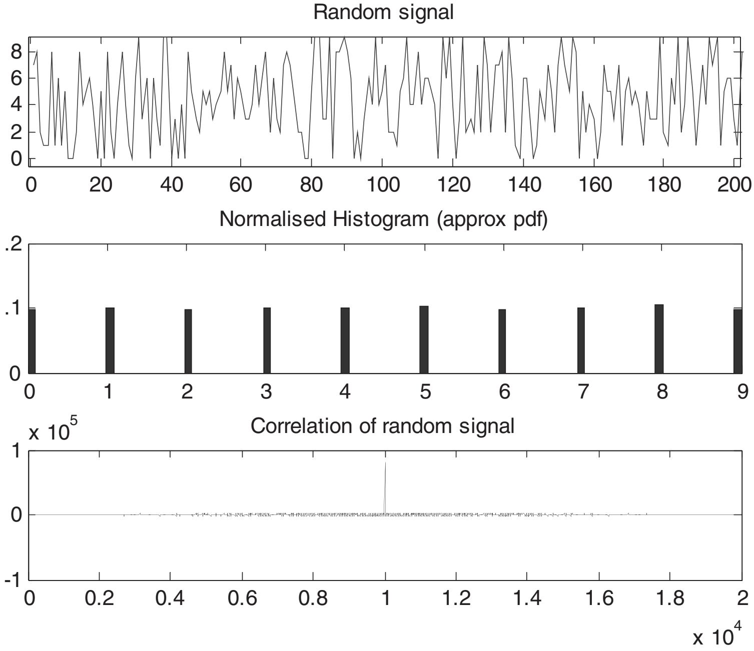


Figure 3.5 Illustration of a uniformly distributed discrete random process with 10 values/states $0, 1, 2, \ldots, 9$. Top: a segment of the process, middle: the normalised histogram of the process, bottom: the autocorrelation of the process. The histogram and autocorrelation were obtained from 10^5 samples.

3.2.3 Stochastic and Random Processes

A random process is any process or function that generates random signals. The term 'stochastic process' is broadly used to describe a random process that generates *sequential* random signals such as a sequence of speech samples, a video sequence, a sequence of noise samples, a sequence of stock market data 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, multinomial process, etc.

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.

3.2.4 The Space of Variations 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 communication network as shown in Figure 3.6. The noise on each telephone line fluctuates randomly with time, and may be denoted as $n(m, s)$, 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 by $N(m) = \{n(m,s)\}$, where $n(m,s)$ denotes a realisation of the noise process $N(m)$ on 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 3.3.9 and 3.3.10, we consider ergodic random processes in which time-averaged statistics, from a single realisation of a process, may be used instead of the ensemble-averaged statistics.

Notation: In this chapter $X(m)$, with upper case X, 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 by $\{x(m,s)\}$, forms the ensemble or the space of the process $X(m)$.

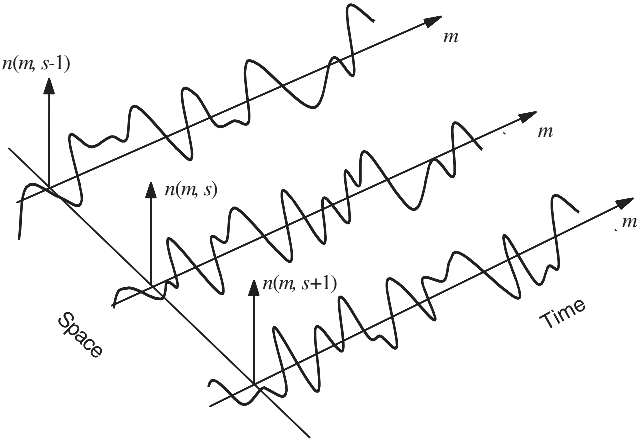


Figure 3.6 Illustration of three different realisations in the space of a random noise process $N(m)$.

3.3 Probability Models of Random Signals

Probability models, devised initially to calculate the odds for the different outcomes in a game of chance, provide a complete mathematical description of the distribution of the likelihood of the different outcomes 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.

3.3.1 Probability as a Numerical Mapping of Belief

It is useful to note that often people quantify their intuitive belief/feeling in the probability of the outcome of a process or a game in terms of a number between zero and one or in terms of its equivalent percentage. A probability of 'zero' expresses the impossibility of the occurrence of an event, i.e. it never happens, whereas a probability of 'one' means that the event is certain to happen, i.e. always happens. Hence a person's belief (perhaps formed by practical experience, intuitive feeling or deductive reasoning) is mapped into a number between one and zero.

3.3.2 The Choice of One and Zero as the Limits of Probability

The choice of zero for the probability of occurrence of an infinitely improbable event, is necessary if the laws of probability are to hold; for example the joint probability of an impossible event *and* a probable event, P (impossible, possible), should be the same as the probability of an impossible event; this requirement can only be satisfied with the use of zero to represent the probability of an impossible event. The choice of one for the probability of an event that happens with certainty is arbitrary but it is a convenient and established choice.

3.3.3 Discrete, Continuous and Finite-State Probability Models

Probability models enable the estimation of the likely values of a process from noisy or incomplete observations. As illustrated in Figure 3.7, probability models can describe random processes that are discrete-valued, continuous-valued or finite-state continuous-valued. Figure 3.7 lists the most commonly used forms of probability models. Probability models are often expressed as functions of the statistical parameters of the random process; most commonly they are in the form of exponential functions of the mean value and covariance of the process.

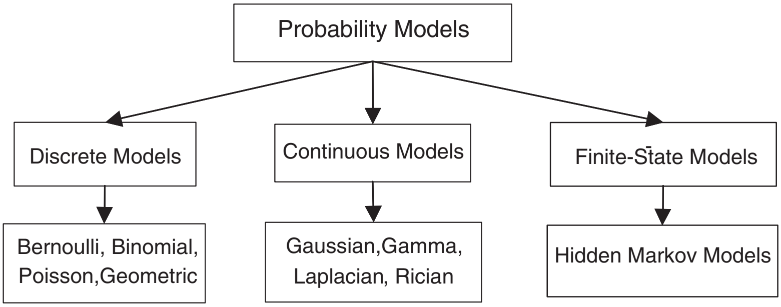


Figure 3.7 A categorisation of different classes and forms of probability models together with some examples in each class.

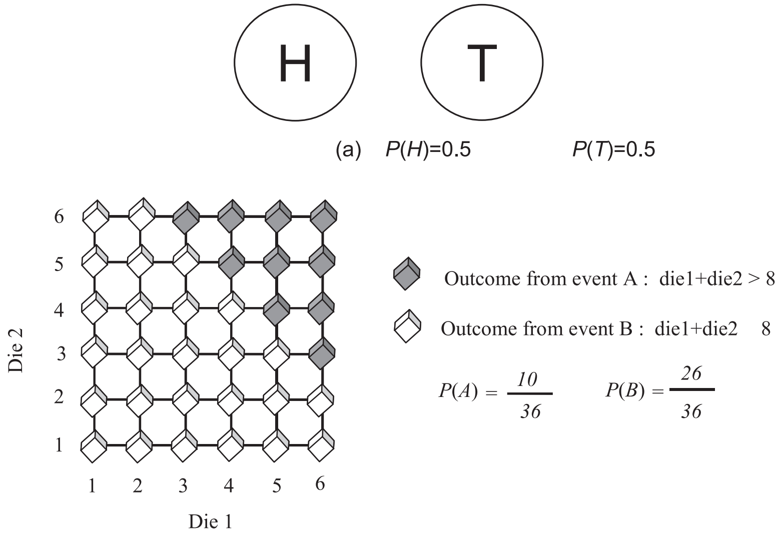
3.3.4 Random Variables and Random Processes

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 time-sequences generated by the same process where m denotes time and s is the sequence index for example as illustrated in Figure 3.6. For a given time instant m, the sample realisation 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.

3.3.5 Probability and Random Variables – The Space and Subspaces of a Variable

Probability models the behaviour of 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 fair coin, Figure 3.8(a), or a pair of fair dice, Figure 3.8(b), or dealing cards in a game.



(a) $P(H)$=0.5 $P(T)$=0.5

(b) The space of two dice.

Figure 3.8 (a) The probability of two outcomes (Head or Tail) of tossing a coin; $P(\mathrm{H}) = P(\mathrm{T}) = 0.5$, (b) 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; $P(A) + P(B) = 1$.

The space of a random variable is the collection of all the values, or outcomes, that the variable can assume. For example the space of the outcomes of a coin is the set {H, T} and the space of the outcome of tossing a die is {1, 2, 3, 4, 5, 6}, Figure 3.8. The space of a random variable can be partitioned, according to some criteria, into a number of subspaces. A subspace is a collection of values with a common attribute, such as the set of outcomes bigger or smaller than a threshold or a cluster of closely spaced samples, or the collection of samples with their values within a given band of values.

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:

$$P(A) = \frac{N_A}{\sum\limits_{\text{All events } i} N_i} \tag{3.5}$$

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

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

Example 3.3

The space of two discrete numbers obtained as outcomes of throwing a pair of dice is shown in Figure 3.8(b). This space can be partitioned in different ways; for example, the two subspaces A and B shown in Figure 3.8 are associated with the pair of numbers that in the case of subspace A add up to a value greater than 8, and in the case of subspace B add up to a value 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.

3.3.6 Probability Mass Function – Discrete Random Variables

For a discrete random variable X that can assume values from a finite set of N numbers or symbols $\{x_1,\ x_2,\ \ldots,\ x_N\}$, each outcome x_i may be considered an event and assigned a probability of occurrence. For example if the variable is the outcome of tossing a coin then the outcomes are Head (H) and Tail (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.

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

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

Bayes' rule forms the foundation of probabilistic estimation and classification. Bayesian inference is introduced in Chapter 4.

Example 3.4 Probability of the sum of two random variables

Figure 3.9(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

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

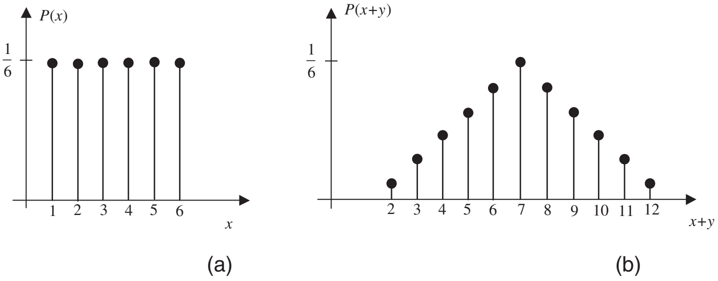


Figure 3.9 The probability mass function (pmf) of (a) a die, and (b) the sum of a pair of dice.

The pmf of the sum of two dice is plotted in Figure 3.9(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. Note that from the central limit theorem, the sum of many independent random variables will have a normal distribution.

3.3.8 Probability Density Function – Continuous Random Variables

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

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

$$F_X(x) = \text{Prob}(X \le x) \tag{3.11}$$

where Prob(·) 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

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

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

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

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

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

The conditional and marginal probability functions and the Bayes' rule, of Equations (3.7)–(3.9), also apply to probability density functions of continuous-valued variables. Figure 3.10 shows the cumulative density function (cdf) and probability density function of a Gaussian variable.

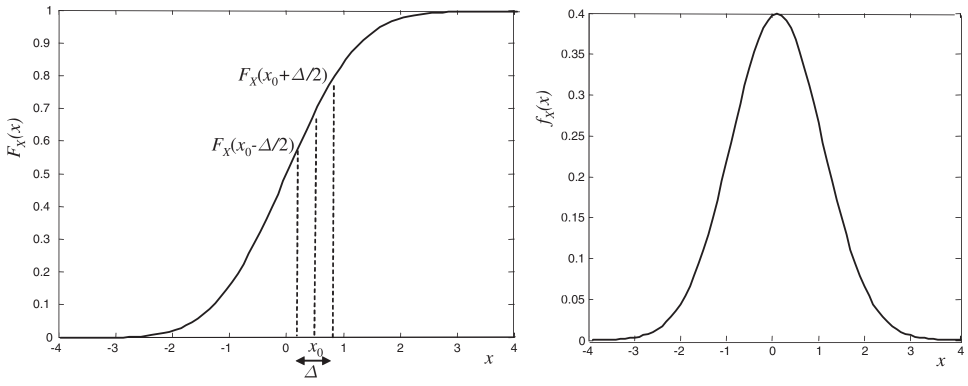


Figure 3.10 The cumulative density function (cdf) $F_X(x)$ and probability density function $f_X(x)$ of a Gaussian variable.

3.3.9 Probability Density Functions of Continuous Random Processes

The probability models obtained for random variables can be applied to random processes such as time series, speech, images, etc. 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 bivariate 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 $f_{X(m_1)X(m_2)\ldots X(m_M)}(x_1,\ x_2,\ \ldots,\ x_M)$ 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

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

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

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

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

$$f_{X(m)|X(n)}\left(x_m\,|x_n\right) = \frac{f_{X(n)|X(m)}\left(x_n\,|x_m\right) f_{X(m)}\left(x_m\right)}{f_{X(n)}\left(x_n\right)} \tag{3.17}$$

Equation (3.17) is 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

$$f_{\left[X(m_1)\cdots X(m_M)\middle|X(n_1)\cdots X(n_N)\right]}\left(x_{m_1},\ldots,\ x_{m_M}\,\middle|x_{n_1},\ldots,\ x_{n_N}\right) = \prod_{i=1}^{M} f_{X(m_i)}(x_{m_i}) \tag{3.18}$$

Discrete-valued random processes can only assume values from a finite set of allowable numbers $[x_1,\ x_2,\ \ldots,\ x_n]$. 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 as

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

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

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

and for a statistically independent process we have

$$P_{[X(m_1)\cdots X(m_M)|X(n_1)\cdots X(n_N)]}\left(x_{m_1},\ldots,\ x_{m_M}\ \middle|\ x_{n_1},\ldots,\ x_{n_N}\right)=\prod_{i=1}^{M}P_{X(m_i)}(X(m_i)=x_{m_i}) \tag{3.21}$$

3.3.10 Histograms – Models of Probability

A histogram is a bar graph that shows the number of times (or the normalised fraction of times) that a random variable takes values in each class or in each interval (bin) of values.

Given a set of observations of a random variable, the range of values of the variable between the minimum value and the maximum value are divided into N equal-width bins and the number of times that the variable falls within each bin is calculated.

A histogram is an estimate of the probability distribution of a variable derived from a set of observations. The MATLAB® routine hist(x, N) displays the histogram of the variable x in N uniform intervals. Figure 3.11 shows the histogram and the probability model of a Gaussian signal. Figure 3.12 shows the scatter plot of a two-dimensional Gaussian process superimposed on an ellipse which represents the standard-deviation contour.

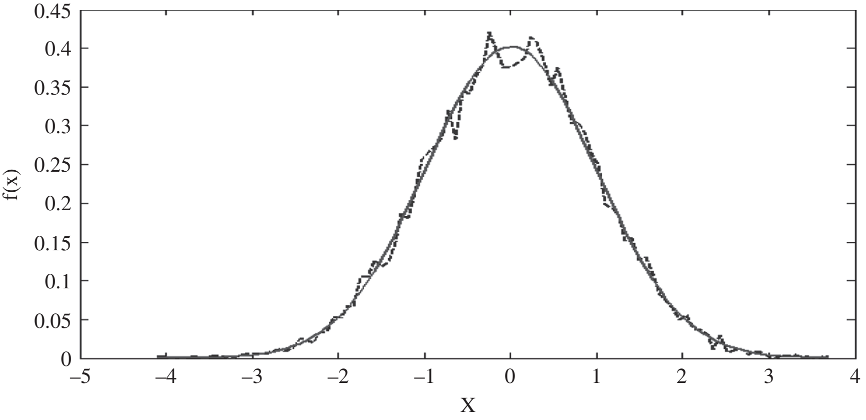


Figure 3.11 Histogram (dashed line) and probability model of a Gaussian signal.

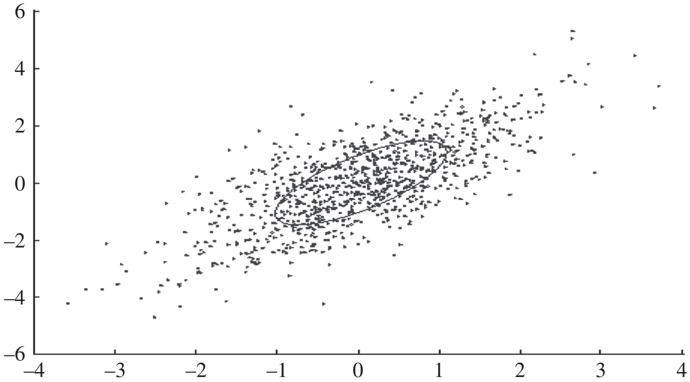


Figure 3.12 The scatter plot of a two-dimensional Gaussian distribution.

3.4 Information Models

As explained, information is knowledge or data about the state sequence of a random process, such as how many states the random process has, how often each state is observed, the outputs of each state, how the process moves across its various states and what is its past, current or likely future states.

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

Information is usually discrete (or quantised) and can be represented in a binary format in terms of M states of a variable. The states of an information-bearing variable may be arranged in a binary tree structure as shown later in Examples 3.10 and 3.11.

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

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

(1) Possess two or more states or values.
(2) Move between the states in a random manner.

For example, the outcome of tossing a coin is an unpredictable binary state (Head/Tail) event, a digital communication system with N-bit code words 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.}.

Since random processes are modelled with probability functions, it is natural that information is also modelled by a function of probability. The negative logarithm of probability of a variable is used as a measure of the information content of the variable. Note that since probability is less than one, the negative log of probability has a positive value. Using this measure an event with a probability of one has zero information whereas the more unlikely an event the more the information it carries when the event is observed.

The *expected* (*average*) *information content* of a state x_i of a random variable is quantified as

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

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

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

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

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

Note from Figure 3.13 that the expected information content of a variable has a value of zero for an event whose probability is zero, i.e. an impossible event, and its value is also zero for an event that happens with probability of one.

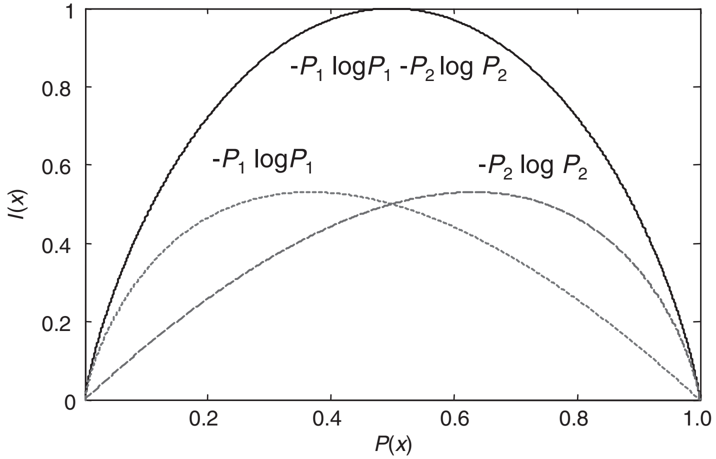


Figure 3.13 Illustration of $I(x_i)$ vs $P(x_i)$; for a binary source, the maximum information content is one bit, when the two states have equal probability of 0.5. $P(x_1) = P_1$ and $P(x_2) = P_2$.

3.4.1 Entropy: A Measure of Information and Uncertainty

Entropy provides 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 is an indicator of the amount of randomness or uncertainty of a discrete random process.

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*, Claude Elwood Shannon derived the entropy information measure, H, as a function that satisfies the following conditions:

(1) Entropy H should be a continuous function of probability 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) subsets, the entropy of the original set should be equal to the probability-weighted sum of the entropy of the subsets.

Consider a random variable X with M states $[x_1, x_2, \ldots, x_M]$ and state probabilities $[p_1, p_2, \ldots, p_M]$ where $P_X(x_i) = p_i$, the entropy of X is defined as

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

where the base of the logarithm is 2. The base 2 of the logarithm reflects the binary nature of information. Information is discrete and can be represented by a set of binary symbols.

$\log_2$ has several useful properties. $\log_2$ of 1 is zero, which is a useful mapping as an event with probability of one has zero information. Furthermore, with the use of logarithm the addition of a binary symbol (with two states) to M existing binary symbols doubles the number of possible outcomes from 2^M to 2^{M+1} but increases the logarithm (to the base 2) of the number of outcomes by one. With the entropy function two equal probability states corresponds to one bit of information, four equal probability states corresponds to two bits and so on. Note that the choice of base 2 gives one bit for a two-state equi-probable random variable.

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

$$0 \le H(X) \le \log_2 M \tag{3.26}$$

where $H(X) = 0$ if one symbol x_i has a probability of one and all other symbols have probabilities of zero, and M denotes the number of symbols in the set X. *The entropy of a set attains a maximum value of* log_2M *bits for a uniformly distributed M-valued variable with each outcome having an equal probability of* $1/M$. Figure 3.14 compares the entropy of two discrete-valued random variables; one is derived from a uniform process, the other is obtained from quantisation of a Gaussian process.

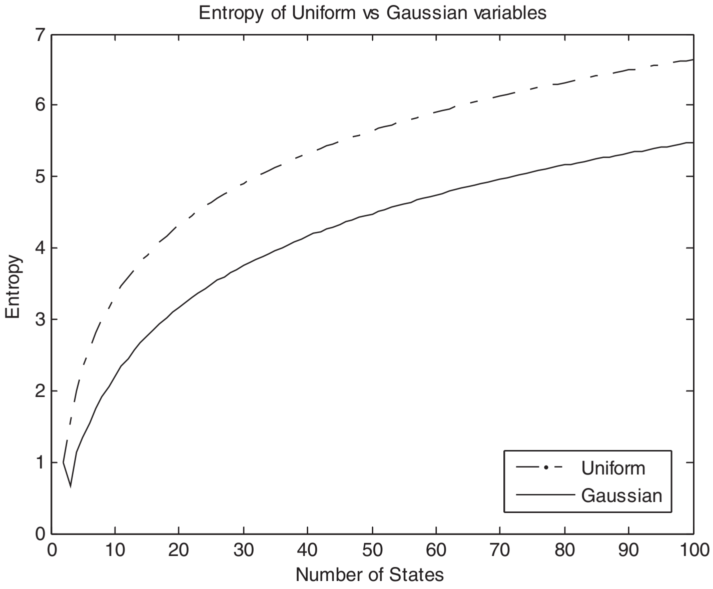


Figure 3.14 Entropy of two discrete random variables as a function of the number of states. One random variable has a uniform probability and the other has a Gaussian-shaped probability function.

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 $K/N \ge H(X)$. As N becomes large, for an efficient code the number of bits per sample K/N approaches the entropy $H(X)$ of X (see Huffman coding).

Shannon's source coding theorem states: 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 $N \to \infty$.

Example 3.5 Entropy of the English alphabet

Calculate the entropy of the set letters in 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 5 letters per word.

Solution:

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

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

The total number of bits for encoding 2000 words = 4.7 × 2000 × 5 = 47 kbits. Note that different letter type cases (upper case, lower case, etc), font types (bold, italic, 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 the alphabet shown in Figure 3.15: $P(\text{A}) = 0.0856$, $P(\text{B}) = 0.0139$, $P(\text{C}) = 0.0279$, $P(\text{D}) = 0.0378$, $P(\text{E}) = 0.1304$, $P(\text{F}) = 0.0289$, $P(\text{G}) = 0.0199$, $P(\text{H}) = 0.0528$, $P(\text{I}) = 0.0627$, $P(\text{J}) = 0.0013$, $P(\text{K}) = 0.0042$, $P(\text{L}) = 0.0339$, $P(\text{M}) = 0.0249$, $P(\text{N}) = 0.0707$, $P(\text{O}) = 0.0797$, $P(\text{P}) = 0.0199$, $P(\text{Q}) = 0.0012$, $P(\text{R}) = 0.0677$, $P(\text{S}) = 0.0607$, $P(\text{T}) = 0.1045$, $P(\text{U}) = 0.0249$, $P(\text{V}) = 0.0092$, $P(\text{W}) = 0.0149$, $P(\text{X}) = 0.0017$, $P(\text{Y}) = 0.0199$, $P(\text{Z}) = 0.0008$.

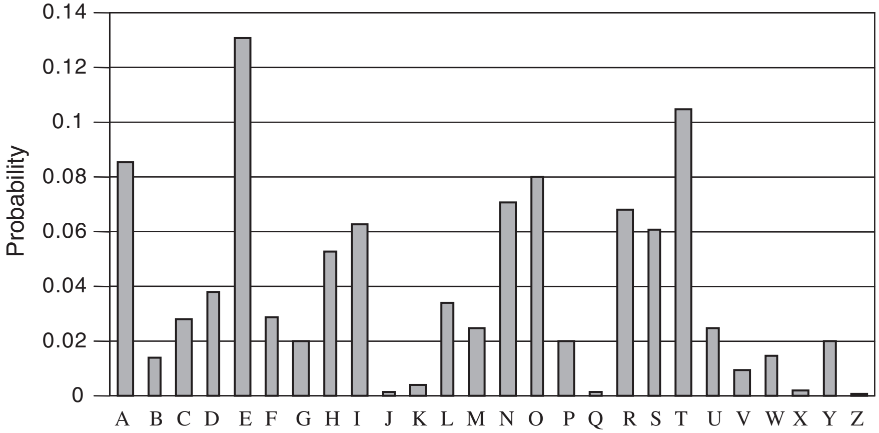


Figure 3.15 The probability (histogram) of the letters of the English alphabet, A to Z.

The entropy of this set is given by

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

Example 3.7 Entropy of the English phonemes

Spoken English is constructed from about 40 basic acoustic symbols, known as phonemes (or phonetic units); 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 equi-probable, and 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.

Solution:

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

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

$$\text{Number of bits/sec} = (120/60 \text{ words per second}) \times (4 \text{ phonemes per word})$$
$$\times (5.3 \text{ bits per phoneme}) = 43.4 \,\text{bps}$$

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 hence the entropy) in speech due to contextual variations of phonemes, speaker identity, accent, pitch intonation and emotion signals.

3.4.2 Mutual Information

Consider two dependent random variables X and Y, the conditional entropy of X given Y is defined as

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

$H(X|Y)$ is equal to $H(X)$ if Y is independent of X and is equal to zero 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

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

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

Example 3.8 Upper and lower bounds on mutual information

Obtain the bounds on the mutual information of two random variables X and Y.

Solution:
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

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

The lower bound is given by the case when X and Y are independent, $P(x_i,\ y_j) = P(x_i)P(y_j)$, i.e. have no mutual information, hence

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

Example 3.9
Show that the *mutual entropy* of two independent variables X and Y are additive.

Solution:
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 $P(x_i,\ y_j) = P(y_i)P(y_j)$ in Equation (3.34) yields

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

where we have used the following relations

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

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 – Variable Length Codes

As explained above, 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 $H(X)/(K/N)$. When $K/N = H(X)$ then the entropy coding efficiency of the code is $H(X)/(K/N) = 1$ or 100 %.

For an information source X the average number of bits per symbol, aka the average code length CL, can be expressed as

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

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 $-\log_2 P(x_i)$. The aim of the design of a 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 $[x_i]$ 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 the 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.

3.4.4 Huffman Coding

A simple and efficient form of entropy coding is the *Huffman code* which creates a set of prefix-free codes (no code is part of the beginning of another code) for a given discrete random source. The ease with which Huffman codes can be created and used makes this code a popular tool for data compression. Huffman devised his code while he was a student at MIT.

In one form of Huffman tree code, illustrated in Figure 3.16, the symbols are arranged in the decreasing order of probabilities in a column. The two symbols with the lowest probabilities are combined by drawing a straight line from each and connecting them. This combination is combined with the next symbol and the procedure is repeated to cover all symbols. Binary code words are assigned by moving from the root of the tree at the right-hand side to left in the tree and assigning a 1 to the lower branch and a 0 to the upper branch where each pair of symbols have been combined.

Example 3.10

Given five symbols $x_1,\ x_2, \ldots,\ x_5$ 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.

Solution:

The entropy of X is $H(X) = 2.122$ bits/symbol. Figure 3.16 illustrates the design of a Huffman code for this source. For this tree we have:

$$\text{average codeword length} = 1 \times 0.4 + 2 \times 0.2 + 4 \times 0.1 + 4 \times 0.1 + 3 \times 0.2 = 2.2\,\text{bits/symbol}$$

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 3.17 illustrates a Huffman code tree created by a series of successive binary divisions 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

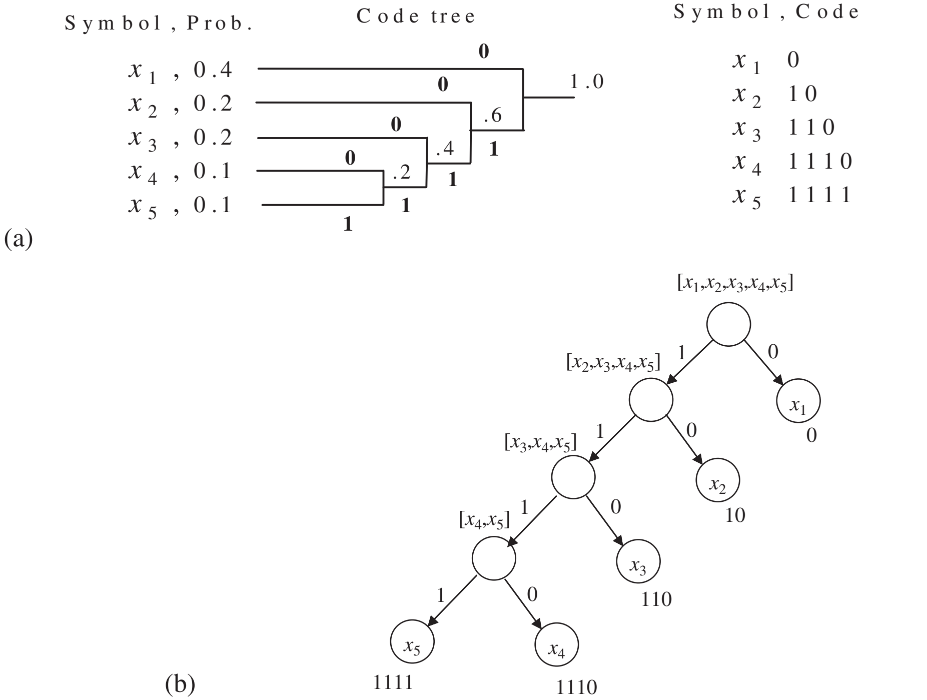


Figure 3.16 (a) Illustration of Huffman coding tree; the source entropy is 2.1219 bits/sample and Huffman code gives 2.2 bits/sample. (b) Alternative illustration of Huffman tree.

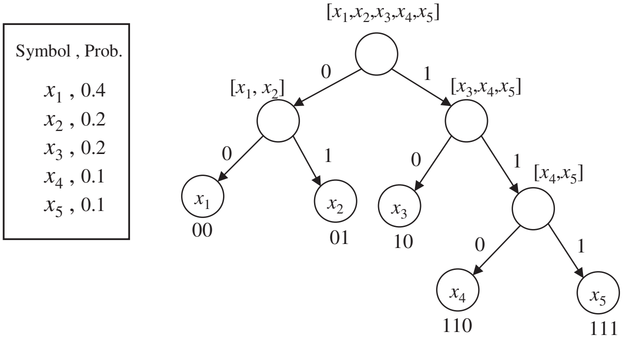


Figure 3.17 A binary Huffman coding tree. From the top node at each stage the set of symbols are divided into two sets with as near set probability (max entropy) as possible. Each end-node with a single symbol represents a leaf and is assigned a binary code which is read from the top node to the leaf node.

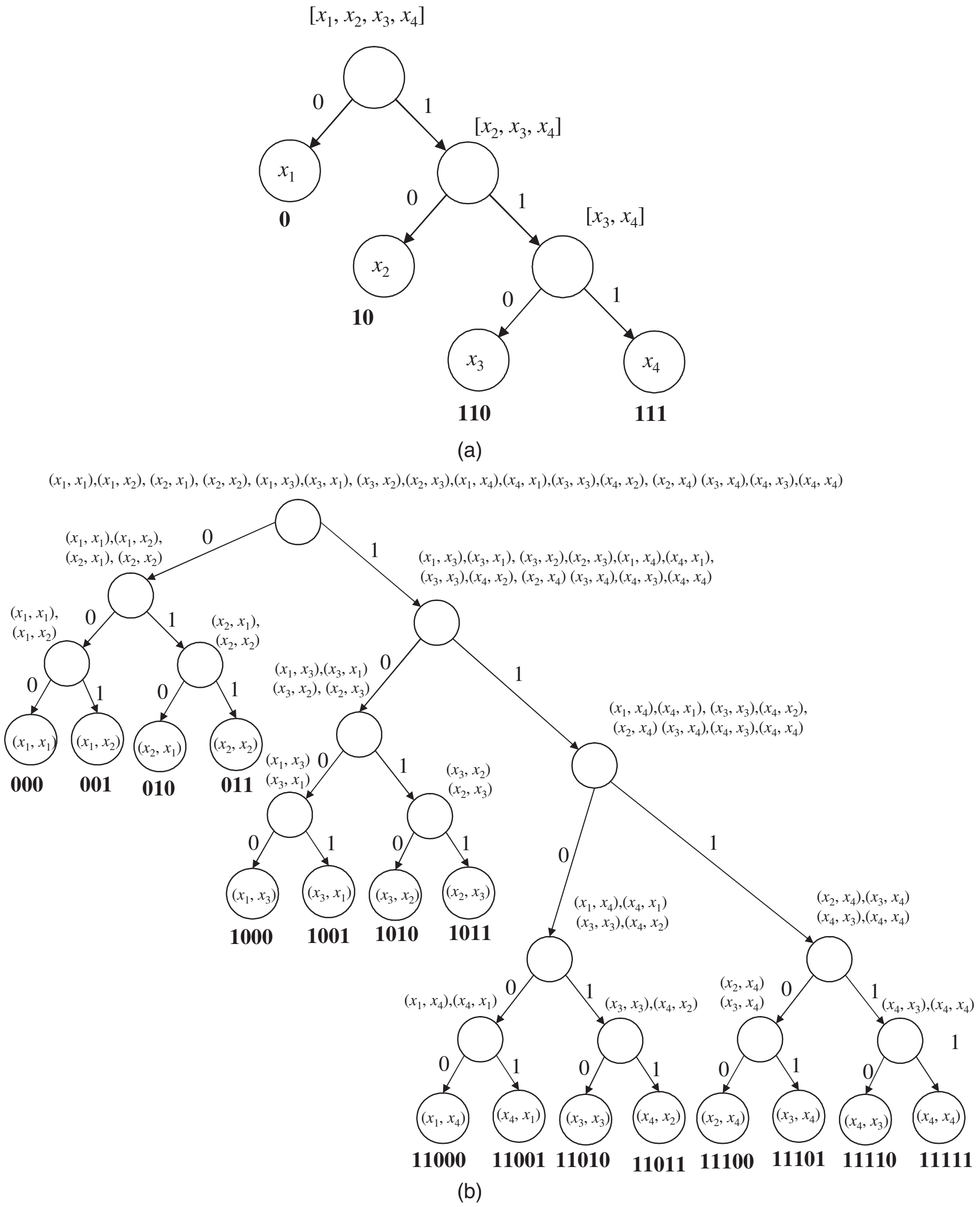
to tree branches as shown. Each end-node with a single symbol represents a leaf node and is assigned a binary code which is read from the top (root) node to the leaf node.

Example 3.11

Given a communication system with four symbols x_1, x_2, x_3 and x_4 and 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.

Solution:

First we note that the entropy of the four symbols is obtained from Equation (3.25) as 1.8464 bits/symbol. Using a Huffman code tree, as illustrated in Figure 3.18(a), the codes are $x_1 \rightarrow 0,\ x_2 \rightarrow 10,\ x_3 \rightarrow 110$ and $x_4 \rightarrow 111$. The average length of this code is 1.9 bits/symbol.



$[x_1, x_2, x_3, x_4]$
0
1
$[x_2, x_3, x_4]$
x_1
0
0
1
$[x_3, x_4]$
x_2
10
0
1
x_3
x_4
110
111
(a)
$(x_1, x_1),(x_1, x_2), (x_2, x_1), (x_2, x_2), (x_1, x_3),(x_3, x_1), (x_3, x_2),(x_2, x_3),(x_1, x_4),(x_4, x_1),(x_3, x_3),(x_4, x_2), (x_2, x_4) (x_3, x_4),(x_4, x_3),(x_4, x_4)$
$(x_1, x_1),(x_1, x_2), (x_2, x_1), (x_2, x_2)$
0
1
$(x_1, x_3),(x_3, x_1), (x_3, x_2),(x_2, x_3),(x_1, x_4),(x_4, x_1), (x_3, x_3),(x_4, x_2), (x_2, x_4) (x_3, x_4),(x_4, x_3),(x_4, x_4)$
$(x_1, x_1), (x_1, x_2)$
0
1
$(x_2, x_1), (x_2, x_2)$
$(x_1, x_3),(x_3, x_1) (x_3, x_2), (x_2, x_3)$
0
1
0
1
0
1
$(x_1, x_4),(x_4, x_1), (x_3, x_3),(x_4, x_2), (x_2, x_4) (x_3, x_4),(x_4, x_3),(x_4, x_4)$
(x_1, x_1)
(x_1, x_2)
(x_2, x_1)
(x_2, x_2)
000
001
010
011
(x_1, x_3) (x_3, x_1)
0
1
(x_3, x_2) (x_2, x_3)
0
1
0
1
0
1
(x_1, x_3)
(x_3, x_1)
(x_3, x_2)
(x_2, x_3)
1000
1001
1010
1011
$(x_1, x_4),(x_4, x_1) (x_3, x_3),(x_4, x_2)$
$(x_2, x_4),(x_3, x_4) (x_4, x_3),(x_4, x_4)$
$(x_1, x_4),(x_4, x_1)$
0
1
$(x_3, x_3),(x_4, x_2)$
$(x_2, x_4) (x_3, x_4)$
0
1
$(x_4, x_3),(x_4, x_4)$
1
0
1
0
1
0
1
0
(x_1, x_4)
(x_4, x_1)
(x_3, x_3)
(x_4, x_2)
(x_2, x_4)
(x_3, x_4)
(x_4, x_3)
(x_4, x_4)
11000
11001
11010
11011
11100
11101
11110
11111
(b)

Figure 3.18 A Huffman coding binary tree for Example 6.11: (a) individual symbols are coded, (b) pairs of symbols are coded.

Assuming the symbols are independent the probability of 16 pairs of symbols can be written as

$$P(x_1,x_1)=0.16, P(x_1,x_2)=0.12, P(x_1,x_3)=0.08, P(x_1,x_4)=0.04$$

$$P(x_2,x_1)=0.12, P(x_2,x_2)=0.09, P(x_2,x_3)=0.06, P(x_2,x_4)=0.03$$

$$P(x_3,x_1)=0.08, P(x_3,x_2)=0.06, P(x_3,x_3)=0.04, P(x_3,x_4)=0.02$$

$$P(x_4,x_1)=0.04, P(x_4,x_2)=0.03, P(x_4,x_3)=0.02, P(x_4,x_4)=0.01$$

The entropy of pairs of symbols is 3.6929 which is exactly twice the entropy of the individual symbols.

The 16 pairs of symbols and their probabilities can be used in a Huffman tree code as illustrated in Figure 3.18(b). The average code length for Huffman coding of pairs of symbols is 1.87 compared with the average code length of 1.9 for Huffman coding of individual symbols.

3.5 Stationary and Non-Stationary Random Processes

Although the amplitude of a signal fluctuates with time m, the parameters of the model or the process that generates the signal may be time-invariant (stationary) or time-varying (non-stationary). Examples of non-stationary processes are speech (Figure 3.19) and music whose loudness and spectral composition change continuously, image and video.

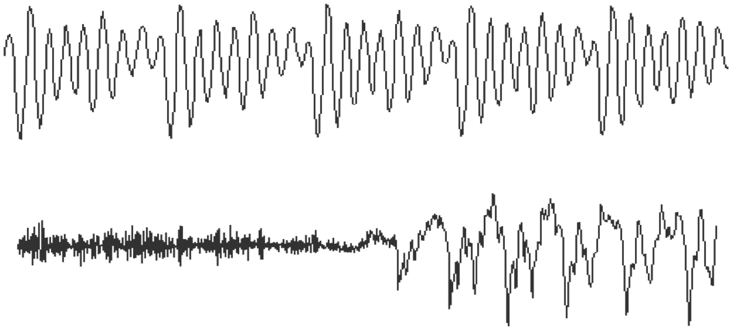


Figure 3.19 Examples of quasi-stationary voiced speech (above) and non-stationary speech composed of unvoiced and voiced speech segments.

A process is stationary if the parameters of the probability model of the process are time-invariant; otherwise it is non-stationary. 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.

Example 3.12

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

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

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

where T is the period of the sine wave.

The mean and the power of the transient signal are given by

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

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

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 non-stationary random process

Consider a non-stationary signal $y(m)$ generated by a binary-state random process, Figure 3.20, described by the equation

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

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

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

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

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

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

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

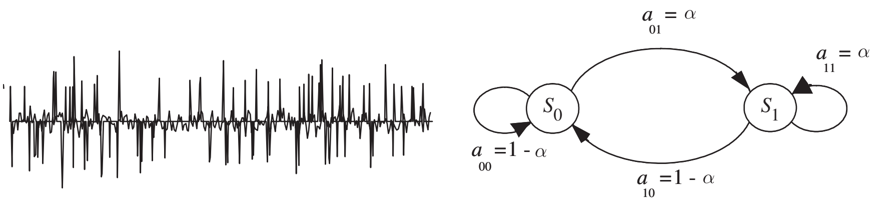


Figure 3.20 Illustration of an impulsive noise signal observed in background noise (left) and a binary-state model of the signal (right).

Although most signals are non-stationary, the concept of a stationary process plays an important role in the development of signal processing methods. Furthermore, most non-stationary 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 strict-sense stationary if all its distributions and statistics are time-invariant. Strict-sense stationarity implies that the n^{th} order distribution is time-invariant (or translation-invariant) for all $n = 1,\ 2,\ 3, \ldots$

$$\begin{aligned} &Prob[x(m_1) \le x_1,\ x(m_2) \le x_2,\ \ldots,\ x(m_n) \le x_n)] \\ &\quad = Prob[x(m_1+\tau) \le x_1,\ x(m_2+\tau) \le x_2, \ldots, x(m_n+\tau) \le x_n)] \end{aligned} \tag{3.49}$$

where τ is any arbitrary shift along the time axis. Equation (3.49) implies that the signal has the same (time-invariant) probability function at all times.

From Equation (3.49) the statistics of a strict-sense stationary process are time invariant. In general for any of the moments of the process we have

$$\begin{aligned} \mathcal{E}\left[x^{k_1}(m_1), x^{k_2}(m_1+\tau_1), \ldots, x^{k_L}(m_1+\tau_L)\right] = \mathcal{E}\left[x^{k_1}(m_2), x^{k_2}(m_2+\tau_2), \ldots,\ x^{k_L}_{t_2+\tau_L}\right. \\ (m_2+\tau_L)] \end{aligned} \tag{3.50}$$

where $k_1, \ldots, k_L$ 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

$$\mathcal{E}\left[x(m)\right] = \mu_x \tag{3.51}$$

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

and

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

where $\mu_x,\ r_{xx}(k)$ 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:

$$\mathcal{E}\left[x(m)\right] = \mu_x \tag{3.54}$$

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

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 Non-Stationary Processes

A random process is a non-stationary process if its statistics vary with time. Most stochastic processes such as video and audio signals, financial data, meteorological data and biomedical signals are non-stationary as they are generated by systems whose contents, environments and parameters vary or evolve over time. For example, speech is a non-stationary 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 3.21. In Figure 3.21(a) a non-stationary process is modelled as the output of a time-varying system whose parameters are controlled by a stationary process. In Figure 3.21(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.

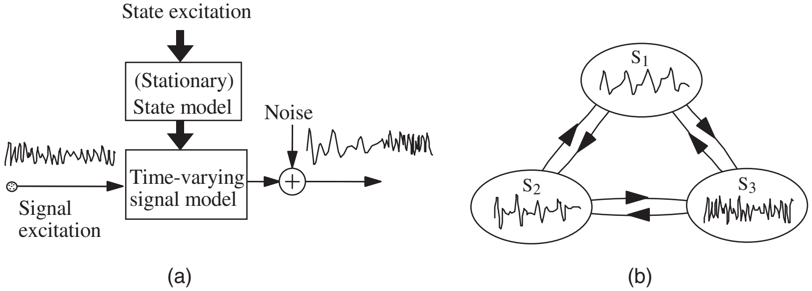


Figure 3.21 Two models for non-stationary processes: (a) a stationary process drives the parameters of a continuously time-varying model; (b) a finite-state model with each state having a different set of statistics.

3.6 Statistics (Expected Values) of a Random Process

The expected values of a random process, also known as its statistics or moments, are the mean, variance, correlation, power spectrum and the higher-order moments 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 its volume 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 use our prior experience or the available data or perhaps our intuitive feeling to select a suitable statistical model for a process, e.g. a Gaussian or Poisson pdf. To complete the model we need to specify the parameters of the model which are usually the expected values of the process such as its mean and covariance. Furthermore, for many algorithms, such as noise reduction filters or linear prediction, what we need essentially 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, $h(X(m_1), X(m_2), \ldots, X(m_M))$, is defined as

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

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 Central Moments

The k^{th} central moment of a random variable x(m) is defined as

$$\mu_k = \mathcal{E}\left[\left((x(m)-\mu(m)\right)^k\right] = \int_{-\infty}^{\infty} \left((x(m)-\mu(m)\right)^k f_{X(m)}(x)\,dx$$

Where m = m1 is the mean value of x(m). Note that the central moments are the moments of random variables about the mean value (i.e. with the mean value removed). The first central moment is zero. The second central moments are the variance and covariance. The third and fourth central moments are skewness and kurtosis respectively.

3.6.1.1 Cumulants

The cumulants are related to expected values of a random variable. The Characteristic function of a random variable is defined as

$$\phi(t) = \mathcal{E}\,[\exp(jtx)] = \int_{-\infty}^{\infty} \exp(jtx)p(x)dx$$

The cumulant generating function is defined as

$$g(t) = \log\phi(t) = \sum_{n=1}^{\infty} k_n \frac{(jt)^n}{n\,!} = k_1(jt) + k_2\frac{(jt)^2}{2} + \cdots$$

where the cumulant function has been expressed in terms of its Taylor series expansion. The n^{th} cumulant is obtained as $k_n = g^{(n)}(0)$, i.e. from evaluation of the n^{th} derivative of the generating function, $g^{(n)}(t)$, at $t = 0$. Hence

$k_1 = g'(0) = \mu$, the mean value
$k_2 = g''(0) = \sigma^2$, the variance value

and so on. Cumulants are covered in more detail in Chapters 16 and 18.

3.6.2 The Mean (or Average) Value

The mean value of a process is its first-order moment. The mean value of a process plays an important part in signal processing and parameter estimation from noisy observations. For example, the maximum likelihood linear estimate of a Gaussian signal observed in additive Gaussian noise is a weighted interpolation between the mean value and the observed value of the noisy signal. The mean value of a vector process $[X(m_1), \ldots, X(m_M)]$ is its average value across the space of the process defined as

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

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

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

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.

Figure 3.22 shows the histograms of the mean and the variance of a random process obtained from 10 000 segments for a uniform process. Each segment is 1000 samples long. Note that from the central limit theorem the estimates of the mean and variance have Gaussian distributions.

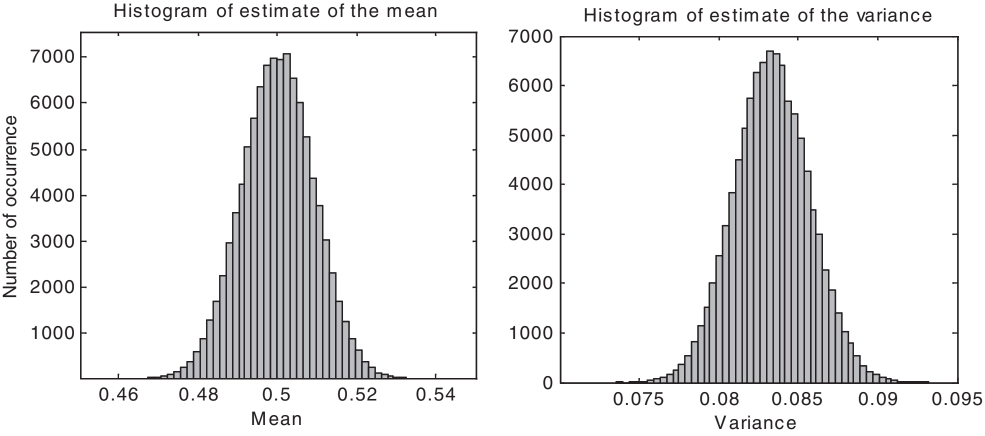
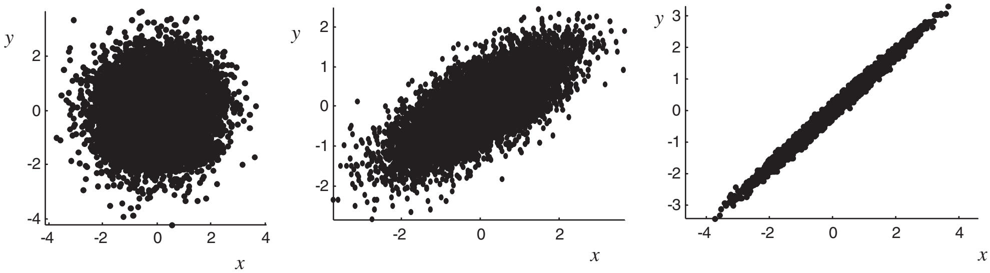


Figure 3.22 The histogram of the estimates of the mean and variance of a uniform random process.

3.6.3 Correlation, Similarity and Dependency

The correlation is a second-order moment or statistic. The correlation of two signals is a measure of the similarity or the dependency of their fluctuations in time or space. Figure 3.23 shows the scatter diagram of two random variables for different values of their correlation.



(a) correlation=0

(b) correlation=0.715

(b) correlation=0.99

Figure 3.23 The scatter plot of two random signals x and y for different values of their cross-correlation. Note that when two variables are uncorrelated their scatter diagram is a circle. As the correlation approaches one the scatter diagram approaches a line.

The values or the courses of action of two correlated signals can be at least partially predicted from each other. Two independent signals have zero correlation. However, two dependent non-Gaussian signals may have zero correlation but non-zero higher order moments, as shown in Chapter 18 on independent component analysis.

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 detectors, digital decoders, predictive coders, digital equalisers, delay estimators, classifiers and signal restoration systems. In Chapter 9 the use of correlations in principal component analysis of speech and image is described.

The correlation of a signal with a delayed version of itself is known as the *autocorrelation*. The autocorrelation function of a random process $X(m)$, denoted by $r_{xx}(m_1, m_2)$, is defined as

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

The autocorrelation function $r_{xx}(m_1, m_2)$ is a measure of the self-similarity, dependency, 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 $r_{xx}(m_1, m_2) = 0$. White noise is an example of such an uncorrelated signal.

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

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

where $k = m_1 - m_2$ is the *autocorrelation lag*.

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

$$r_{xx}(-k) = r_{xx}(k) \tag{3.61}$$

$$r_{xx}(k) \le r_{xx}(0) \tag{3.62}$$

Equation (3.61) says that $x(m)$ has the same relation with $x(m + k)$ as with $x(m - k)$.

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

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

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

$$r_{xx}(k) = \frac{1}{N}\sum_{l=0}^{N-1} |X(l)|^2 e^{j2\pi kl/N} \tag{3.64}$$

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 heartbeat pulses shown in Figure 3.24(a). Figures 3.24(b) and (c) show the estimate of the periods and the autocorrelation function of the signal in Figure 3.24(a) respectively. Note that the 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.

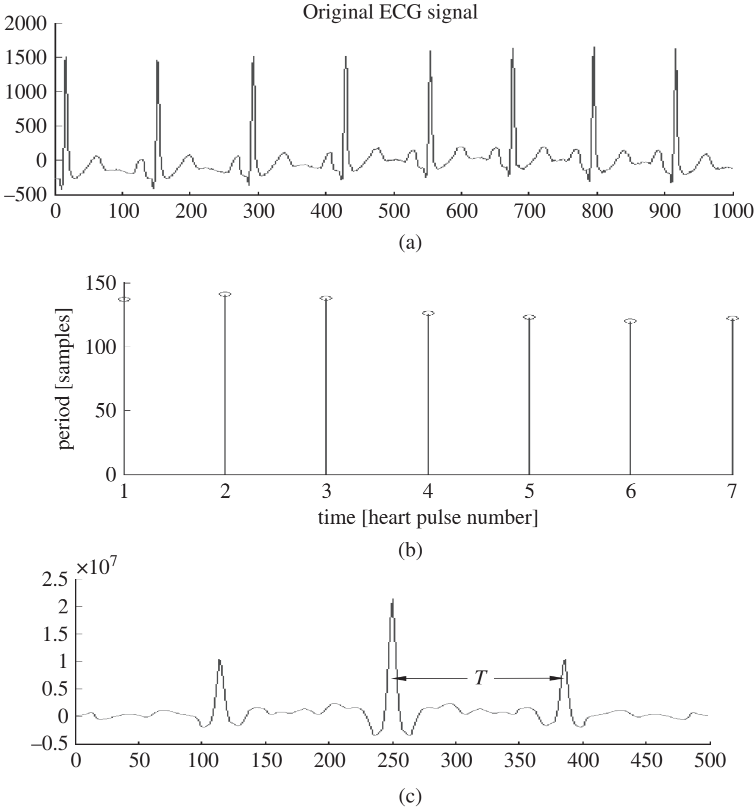


Figure 3.24 (a) Heartbeat signal, electrocardiograph ECG; (b) variation of period with time; (c) autocorrelation function of ECG.

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 a LTI system respectively. The input–output relation is given by

$$y(m) = \sum_i h(i)x(m-i) \tag{3.65}$$

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

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

3.6.4 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

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

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

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

3.6.5 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

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

where $r_{xx}(k)$ 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 a symmetric function, and the power spectrum may be written as

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

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

$$r_{xx}(k) = \int_{-1/2}^{1/2} P_{xx}(f)e^{j2\pi fk}\,df \tag{3.73}$$

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

Example 3.16 Power spectrum and autocorrelation of white noise

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

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

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 $r_{nn}(k)$:

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

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 either in the time domain or in the frequency domain. Figure 3.25 illustrates the shapes of the autocorrelation and power spectrum of a white noise.

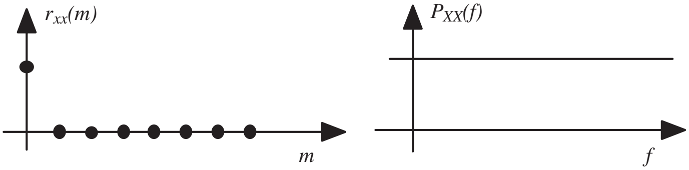


Figure 3.25 Autocorrelation and power spectrum of white noise.

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

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

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

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

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 times of occurrence. A random impulsive noise sequence $n_i(m)$ can be modelled as an amplitude-modulated random binary sequence as

$$n_i(m) = n(m)\; b(m) \tag{3.78}$$

where $b(m)$ is a binary-state random sequence that indicates the presence or the 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

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

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. When $b(m) = 1$, impulsive noise is present and its autocorrelation is equal to $\sigma_n^2\delta(k)$; when $b(m) = 0$, impulsive noise is absent and its autocorrelation is equal to 0. The power spectrum of an impulsive noise sequence is obtained by taking the Fourier transform of the autocorrelation function as

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

3.6.6 Joint Statistical Averages of Two Random Processes

In many signal processing problems, for example in the processing of the outputs of an array of radar sensors, or in smart antenna arrays for mobile phones, we have two or more random processes which may or may not be independent. Joint statistics and joint distributions are used to describe the statistical relationship between two or more random processes.

For two discrete-time random processes $x(m)$ and $y(m)$, the joint pdf is denoted by

$$f_{X(m_1)\cdots X(m_M),Y(n_1)\cdots Y(n_N)}(x_1,\ldots,x_M,\ y_1,\ldots,y_N) \tag{3.81}$$

The joint probability gives the likelihood of two or more variables assuming certain states or values. 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

$$\begin{aligned} f_{X(m_1)\cdots X(m_M),Y(n_1)\cdots Y(n_N)}(x_1,\ldots,x_M,y_1,\ldots,y_N) = f_{X(m_1)\cdots X(m_M)}(x_1,\ldots,x_M)\times \\ f_{Y(n_1)\cdots Y(n_N)}(y_1,\ \ldots,\ y_N) \end{aligned} \tag{3.82}$$

3.6.7 Cross-Correlation and Cross-Covariance

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

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

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

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

where $k = m_1 - m_2$ is the *cross-correlation lag*.

The cross-covariance function is defined as

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

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

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

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

$$y_1(m) = x(m) + n_1(m) \tag{3.87}$$

$$y_2(m) = Ax(m - D) + n_2(m) \tag{3.88}$$

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

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

Assuming that the signal and noise are uncorrelated, we have $r_{y_1y_2}(k) = Ar_{xx}(k - D)$. As shown in Figure 3.26, the cross-correlation function has its maximum at the lag D.

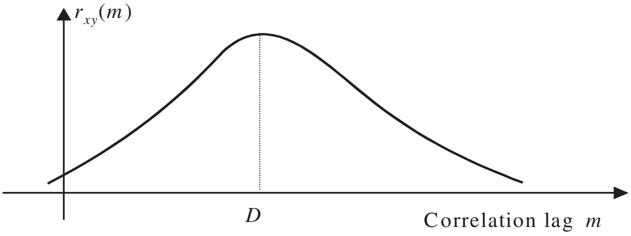


Figure 3.26 The peak of the cross-correlation of two delayed signals can be used to estimate the time delay D.

3.6.8 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:

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

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

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

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

3.6.9 Ergodic Processes and Time-Averaged Statistics

Ergodic theory has its origins in the work of Boltzmann in statistical mechanics problems where time- and space-distribution averages are equal.

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

$$\underset{\text{along time } m}{\text{statistical averages}}[x(m,s)] = \underset{\text{across space } s}{\text{statistical averages}}[x(m,s)] \tag{3.92}$$

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

3.6.10 Mean-Ergodic Processes

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

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

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

$$\lim_{N\to\infty} \mathcal{E}[\hat{\mu}_X] = \mu_X \tag{3.94}$$

$$\lim_{N\to\infty} \text{var}[\hat{\mu}_X] = 0 \tag{3.95}$$

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 it is 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

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

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

$$\begin{aligned}\text{Var}[\hat{\mu}_x] &= \mathcal{E}[\hat{\mu}_x^2] - \mathcal{E}^2[\hat{\mu}_x]\\ &= \mathcal{E}[\hat{\mu}_x^2] - \mu_x^2\end{aligned} \tag{3.97}$$

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

$$\begin{aligned}\mathcal{E}\left[\hat{\mu}_x^2\right] &= \mathcal{E}\left[\left(\frac{1}{N}\sum_{m_1=0}^{N-1} x(m_1)\right)\left(\frac{1}{N}\sum_{m_2=0}^{N-1} x(m_2)\right)\right] \\ &= \frac{1}{N}\sum_{k=-(N-1)}^{N-1}\left(1-\frac{|k|}{N}\right) r_{xx}(k)\end{aligned} \tag{3.98}$$

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

$$\begin{aligned}\mathrm{Var}\left[\hat{\mu}_x^2\right] &= \frac{1}{N}\sum_{k=-(N-1)}^{N-1}\left(1-\frac{|k|}{N}\right) r_{xx}(k) - \mu_x^2 \\ &= \frac{1}{N}\sum_{k=-(N-1)}^{N-1}\left(1-\frac{|k|}{N}\right) c_{xx}(k)\end{aligned} \tag{3.99}$$

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

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=-(N-1)}^{N-1}\left(1-\frac{|k|}{N}\right) c_{xx}(k) = 0 \tag{3.100}$$

3.6.11 Correlation-Ergodic Processes

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

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

The estimate of autocorrelation $\hat{r}_{xx}(k)$ 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

$$\lim_{N\to\infty}\mathcal{E}\left[\hat{r}_{xx}(k)\right] = r_{xx}(k) \tag{3.102}$$

$$\lim_{N\to\infty}\mathrm{Var}\left[\hat{r}_{xx}(k)\right] = 0 \tag{3.103}$$

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

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

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

$$\mathrm{Var}\left[\hat{r}_{xx}(k)\right] = \mathcal{E}\left[\hat{r}_{xx}^2(k)\right] - r_{xx}^2(k) \tag{3.105}$$

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

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

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

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

3.7 Some Useful Practical 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 modelling traffic, 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 in communication systems.
(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:

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

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 $N\ (x,\ \mu_x,\ \sigma_x^2)$. The maximum of a Gaussian pdf occurs at the mean μ_x, and is given by

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

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

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

Figure 3.27 shows the bell-shaped pdf and the cdf of a Gaussian model. The most probable values of a Gaussian process happen around the mean value and the probability of a value decreases exponentially with the increasing distance from the mean value. The total area under the pdf curve is one. 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 of the mean value ($\mu \pm 2\sigma$) is 0.955 and the area three standard deviations on each side of the mean value ($\mu \pm 3\sigma$) is 0.997.

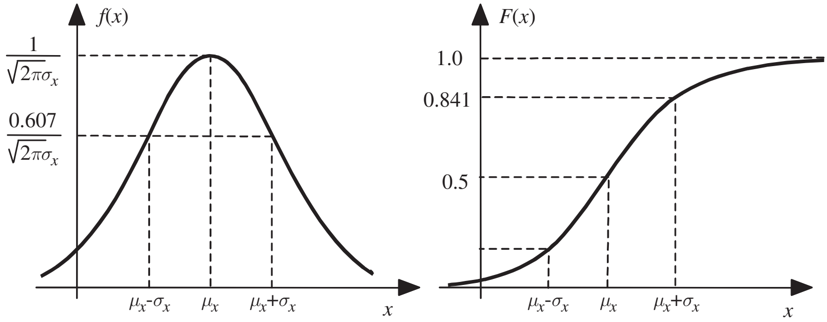


Figure 3.27 Gaussian probability density and cumulative density functions.

3.7.2 Multivariate Gaussian Process

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

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

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

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mathcal{E}[x(m_0)] \\ \mathcal{E}[x(m_2)] \\ \vdots \\ \mathcal{E}[x(m_{P-1})] \end{bmatrix} \tag{3.112}$$

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

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

where $c(m_i,\ m_j)$ is the covariance of elements m_i and m_j of the vector process. The Gaussian process of Equation (3.111) is also denoted by $N(\boldsymbol{x},\ \boldsymbol{\mu}_x,\ \boldsymbol{\Sigma}_{xx})$. 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:

$$f_X\left(\boldsymbol{x} = [x(m_0), \ldots,\ x(m_{P-1})]^{\mathrm{T}}\right) = \prod_{i=0}^{P-1} \frac{1}{\sqrt{2\pi}\ \sigma_{xi}} \exp\left\{-\frac{[x(m_i) - \mu_{xi}]^2}{2\sigma_{xi}^2}\right\} \tag{3.114}$$

Example 3.20 Conditional multivariate Gaussian probability density function

This is useful in situations when we wish to estimate the expectation of vector $\boldsymbol{x}(m)$ given a related vector $\boldsymbol{y}(m)$.

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

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

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

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

$$f_{X|Y}\left(\boldsymbol{x}(m)\ |\boldsymbol{y}(m)\right) = \frac{f_{X,Y}\left(\boldsymbol{x}(m), \boldsymbol{y}(m)\right)}{f_Y\left(\boldsymbol{y}(m)\right)} \tag{3.117}$$

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

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

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

3.7.3 Gaussian Mixture 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

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

where $\mathcal{N}_i(\boldsymbol{x}, \boldsymbol{\mu}_{xi}, \boldsymbol{\Sigma}_{xxi})$ is a multivariate Gaussian density with mean vector $\boldsymbol{\mu}_{xi}$ and covariance matrix $\boldsymbol{\Sigma}_{xxi}$, 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

$$P_i = \frac{N_i}{\sum\limits_{j=1}^{M} N_j} \tag{3.121}$$

where N_i is the number of observations of the process associated with the mixture i. Figure 3.28 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.

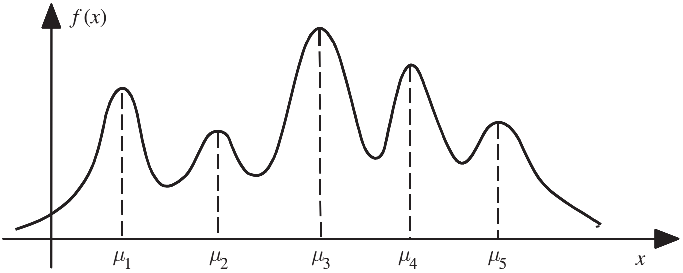


Figure 3.28 A Gaussian mixture model (GMM) pdf.

3.7.4 Binary-State Gaussian Process

A simple example of a binary-state process is the observations at the output of a 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 $\sigma^2_{x,0}$, and in the state s_1 the process is also Gaussian with mean $\mu_{x,1}$ and variance $\sigma^2_{x,1}$. The state-dependent pdf of $x(m)$ can be expressed as

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

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

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

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

$$\sum_S f_{X,\,S}\big(x(m),s_i\big)=f_X\big(x(m)\big) \tag{3.124}$$

$$\int_X f_{X,\,S}\big(x(m),s_i\big)\;dx=P_S(s_i) \tag{3.125}$$

and

$$\sum_S\int_X f_{X,\,S}\big(x(m),s_i\big)\;dx=1 \tag{3.126}$$

Note that in a multi-state 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 multi-state models with a mixture Gaussian pdf per state.

3.7.5 Poisson Process – Counting Process

The Poisson process is a continuous-time integer-valued counting process, used for modelling the probability of the number of 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 distribution 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 $\underset{.}{\lambda} t)$, such that the probability that an event occurs in a small time interval Δt is

$$\text{Prob}\left(1\ \textit{occurrence in the interval}\ (t, t+\Delta t)\right) = \lambda(t)\Delta t \tag{3.127}$$

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 of Δt is given by

$$\text{Prob}\left(0\ \text{occurrence in the interval}\ (t, t+\Delta t)\right) = 1 - \lambda(t)\Delta t \tag{3.128}$$

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 of $t + \Delta t$, denoted by $P\big(k,\ (0,\ t+\Delta t)\big)$:

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

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

$$\lim_{\Delta t \to 0} \frac{P(k, (0, t+\Delta t)) - P(k, (0,t))}{\Delta t} = \frac{dP(k,t)}{dt} = -\lambda P(k,t) + \lambda P(k-1, t) \tag{3.130}$$

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

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

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

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

Figure 3.29 illustrates Poisson pdf. From Equation (3.132), it is easy to show that for a homogeneous Poisson process, the probability of k occurrences of an event in a time interval (t_1, t_2) is given by

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

Equation (3.133) states the probability of k events in a time interval of $t_2 - t_1$, in terms of the rate at which the process happens.

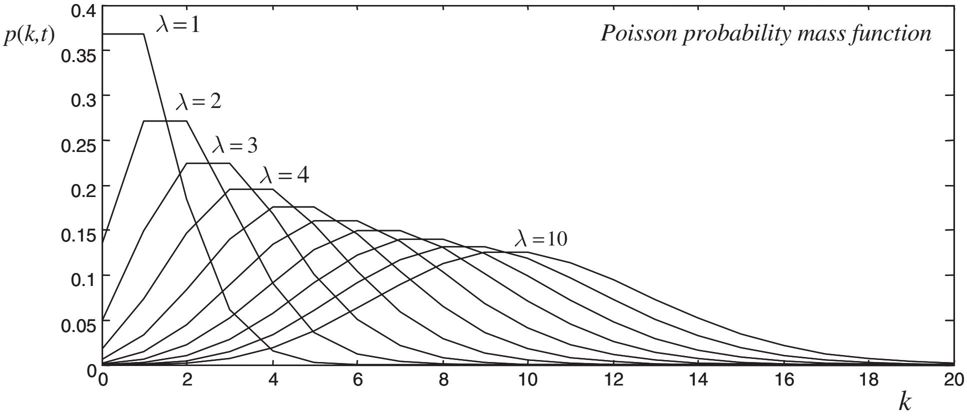


Figure 3.29 Illustration of Poisson process with increasing rate λ from 1 to 10. At each curve the maximum happens at the expected number of events i.e. rate × time interval. Note the probability exists for discrete values of k only.

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

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

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

$$\mathrm{Var}\;[X(t)] = \mathcal{E}\;\lfloor X^2(t)\rfloor\; - \mathcal{E}^2\,[X(t)] = \lambda t \tag{3.136}$$

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 3.30).

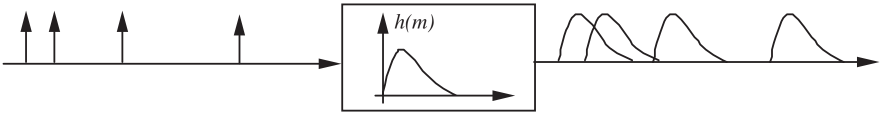


Figure 3.30 Shot noise is modelled as the output of a filter excited with a process.

Consider a Poisson-distributed binary-valued impulse process $x(t)$. Divide the time axis into uniform short intervals of 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

$$\mathcal{E}\lfloor x(m\Delta t)\rfloor = 1 \times P\big(x(m\Delta t) = 1\big)\; + 0 \times\; P\big(x(m\Delta t) = 0\big) = \lambda\Delta t \tag{3.137}$$

and

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

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 $x(t)$:

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

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

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

and

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

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

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

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 a linear filter. The Poisson–Gaussian impulsive noise model is given by

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

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

$$\begin{aligned} f_X\big(x(m)=x_m \ |x(m-1)=x_{m-1},\ldots,x(m-N)=x_{m-N}\big) \\ =f_X\big(x(m)=x_m \ |x(m-1)=x_{m-1}\big) \end{aligned} \tag{3.144}$$

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

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

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 auto-regressive process (Figure 3.31) defined as

$$x(m)=a\,x(m-1)+e(m) \tag{3.146}$$

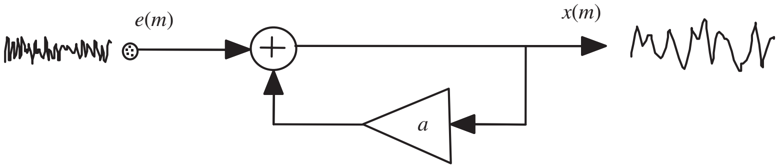


Figure 3.31 A first-order autoregressive (Markov) process; in this model the value of the process at time m, $x(m)$ depends only on $x(m-1)$ and a radon input.

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

$$\begin{aligned} f_X\big(x(m)\,|\,x(m-1),\ldots,x(m-N)\big)=f_X\big(x(m)\,|\,x(m-1)\big) \\ =f_E\big(e(m)=x(m)-\ ax(m-1)\big) \end{aligned} \tag{3.147}$$

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

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

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 3.32). 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

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

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 (3.149), the transition probability is expressed in a general time-dependent form. The marginal probability that a Markov process is in state j at time m, $P_j(m)$, can be expressed as

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

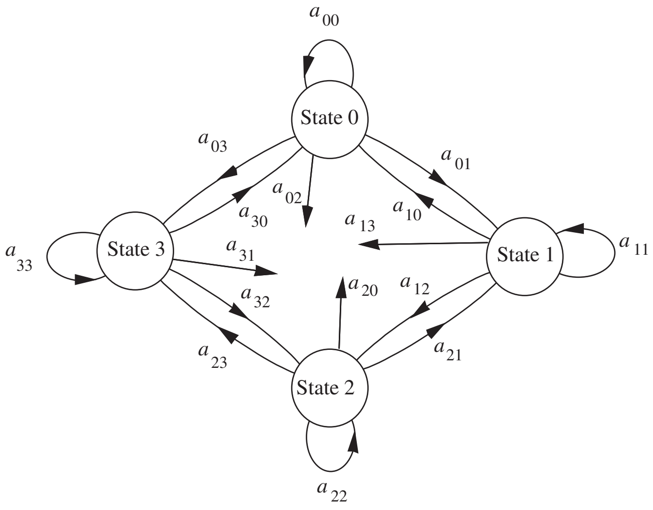


Figure 3.32 A Markov chain model of a four-state discrete-time Markov process.

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

- number of states, N
- state probability vector

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

- and the state transition matrix.

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

3.7.10 Homogeneous and Inhomogeneous Markov Chains

A Markov chain with time-invariant state transition probabilities is known as a homogeneous Markov chain. For a homogeneous 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 equation

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

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

3.7.11 Gamma Probability Distribution

The Gamma pdf is defined as

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

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

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

Note that $\Gamma(1)=1$. The Gamma pdf is sometimes used in modelling speech and image signals. Figure 3.33 illustrates the Gamma pdf.

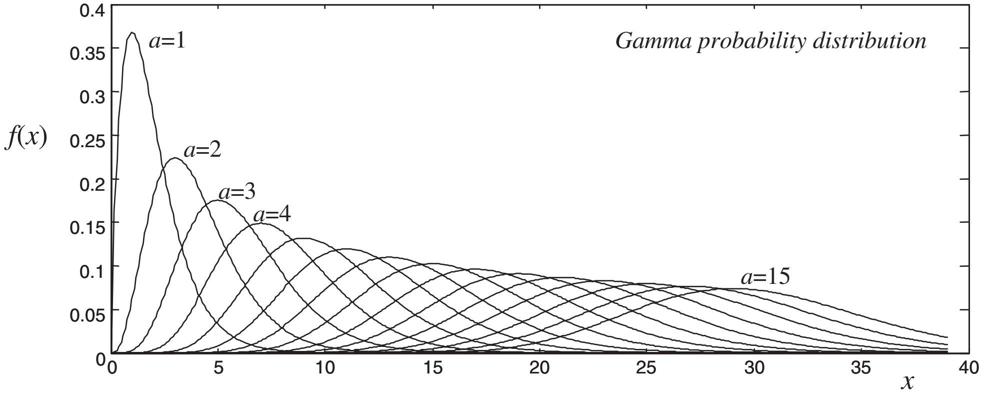


Figure 3.33 Illustration of Gamma pdf with increasing value of a, $b=1$.

3.7.12 Rayleigh Probability Distribution

The Rayleigh pdf, show in Figure 3.34, is defined as

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

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

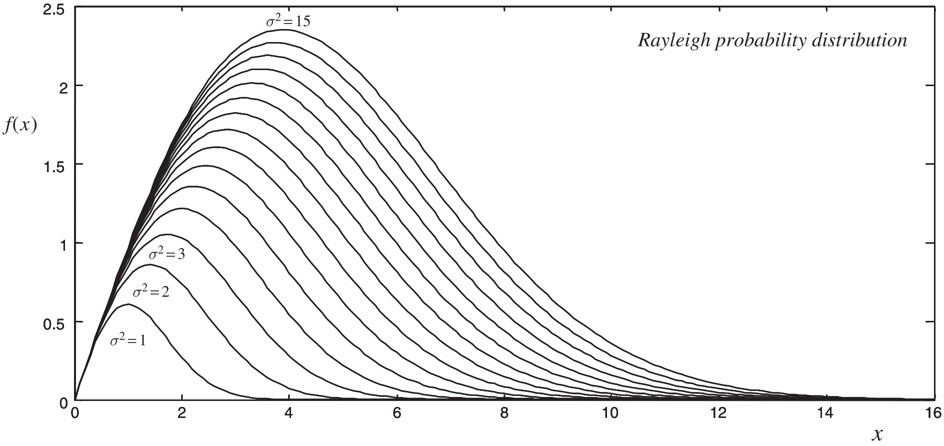


Figure 3.34 Illustration of Rayleigh process with increasing variance σ^2 from 1 to 15.

3.7.13 Chi Distribution

A Chi pdf, Figure 3.35, is defined as

$$p_k(x) = \frac{2^{1-k/2} x^{k-1}}{\Gamma(k/2)} \exp\left(-\frac{x^2}{2}\right) \tag{3.155}$$

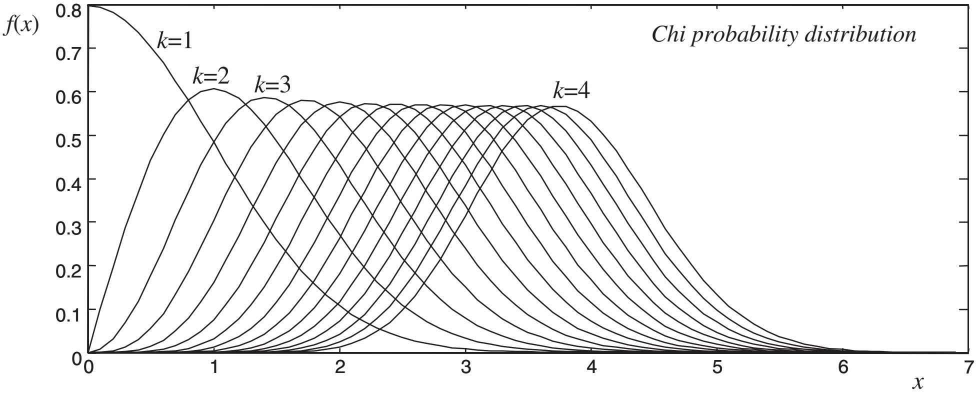


Figure 3.35 Illustration of a Chi process $p_k(x)$ with increasing value of variable k.

As shown in Figure 3.35 the parameter k controls the shape of the distribution. For $k = 2$ the Chi distribution is the same as the Rayleigh distribution (Equation (3.154)) with a standard deviation of $\sigma = 1$.

3.7.14 Laplacian Probability Distribution

A Laplacian pdf, Figure 3.36, is defined as

$$p(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right) \tag{3.156}$$

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

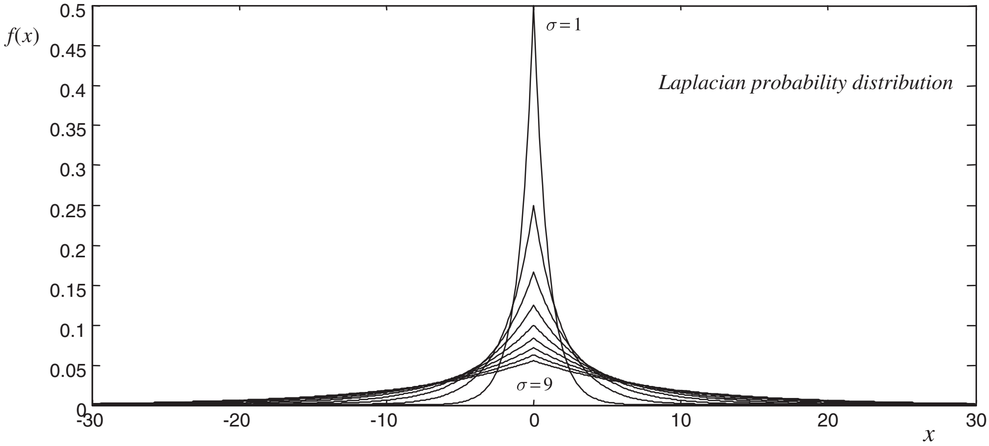


Figure 3.36 Illustration of Laplacian pdfs with standard deviation from 1 to 9.

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 3.37 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 $x(m) \in \{x_1, \ldots, x_N\}$ and $y(m) \in \{y_1, \ldots, y_M\}$ then we have

$$P_Y\big(y(m) = y_j\big) = \sum_{x_i \to y_j} P_X\,(x(m) = x_i) \tag{3.157}$$

where the summation is taken over all values of $x(m)$ that map to $y(m) = y_j$. An example of discrete-valued mapping is quantisation from a larger codebook to a smaller codebook. 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

$$\text{Prob}[y(m) < Y < y(m) + \Delta y] = \int_{x(m)|y(m)<Y<y(m)+\Delta y} f_X\big(x(m)\big)\, dx(m) \tag{3.158}$$

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

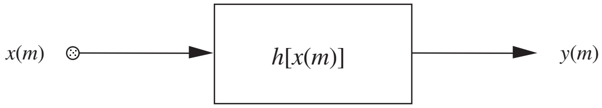


Figure 3.37 Transformation of a random process $x(m)$ to an output process $y(m)$.

3.8.1 Monotonic Transformation of Random Processes

Now for a monotonic one-to-one transformation $y(m) = h[x(m)]$ (e.g. as in Figure 3.38) Equation (3.158) becomes

$$\mathrm{Prob}\big(y(m) < Y < y(m) + \Delta y\big) = \mathrm{Prob}\big(x(m) < X < x(m) + \Delta x\big) \tag{3.159}$$

or, in terms of the cumulative distribution functions

$$F_Y\big(y(m) + \Delta y\big) - F_Y\big(y(m)\big) = F_X\big(x(m) + \Delta x\big) - F_X\big(x(m)\big) \tag{3.160}$$

Multiplication of the left-hand side of Equation (3.160) by $\Delta y / \Delta y$ and the right-hand side by $\Delta x / \Delta x$ and re-arrangement of the terms yields

$$\frac{F_Y\big(y(m) + \Delta y\big) \ - F_Y\big(y(m)\big)}{\Delta y} = \frac{\Delta x}{\Delta y} \frac{F_X\big(x(m) + \Delta x\big) - F_X\big(x(m)\big)}{\Delta x} \tag{3.161}$$

Now as the intervals Δx and Δy tend to zero, Equation (3.161) becomes

$$f_Y\big(y(m)\big) = \left| \frac{\partial x(m)}{\partial y(m)} \right| f_X\big(x(m)\big) \tag{3.162}$$

where $f_Y(y(m))$ is the probability density function. In Equation (3.162), substitution of $x(m) = h^{-1}(y(m))$ yields

$$f_Y\big(y(m)\big) = \left| \frac{\partial\, h^{-1}\big(y(m)\big)}{\partial\, y(m)} \right| f_X\Big(h^{-1}\big(y(m)\big)\Big) \tag{3.163}$$

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

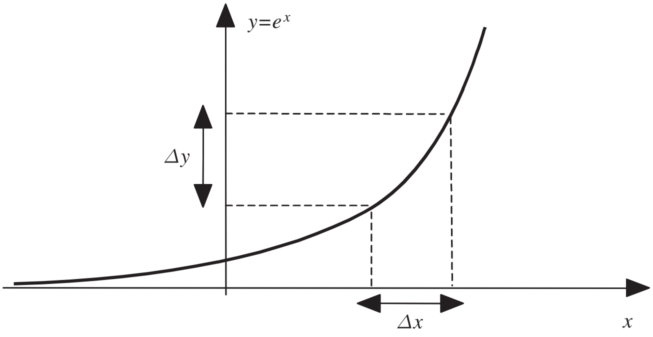


Figure 3.38 An example of a monotonic one-to-one mapping.

Example 3.22 Probability density of a scaled variable

Consider the simple scaled function

$$y(m) = \alpha x(m) \tag{3.164}$$

From Equation (3.163), noting that $h^{-1}\big(y(m)\big) = \frac{1}{\alpha} y(m)$, we have

$$f_Y\big(y(m)\big) = \left|\frac{1}{\alpha}\right| f_X\big(y(m)/\alpha\big) \tag{3.165}$$

Example 3.23 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

$$X(k) = X_R(k) + X_I(k) = r(k)e^{j\varphi(k)} \tag{3.166}$$

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 with a pdf that can be described as

$$\begin{aligned} f_{X_R,\,X_I}(X_R, X_I) = f\,(X_R) f\,(X_I) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{X_R^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{X_I^2}{2\sigma^2}\right) \\ &= \frac{1}{2\pi\sigma^2} \exp\left(-\frac{X_R^2 + X_I^2}{2\sigma^2}\right) \end{aligned} \tag{3.167}$$

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

$$dX_R\, dX_I = r\, dr d\varphi \tag{3.168}$$

Hence from Equation (3.162) we have

$$f_{R,\Phi}(r, \varphi) = \frac{dX_R dX_I}{dr d\varphi} f_{X_R X_I}(X_R, X_I) = r f_{X_R, X_I}(X_R, X_I) \tag{3.169}$$

From Equations (3.167) and (3.169) we have

$$f_{R,\Phi}(r, \varphi) = \frac{r}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) \tag{3.170}$$

The probability of the phase φ can be obtained from

$$f_\Phi(\varphi) = \int_0^\infty f_{R,\Phi}(r, \varphi) dr = \frac{1}{2\pi} \int_0^\infty \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) dr = \frac{1}{2\pi} \tag{3.171}$$

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

$$f_R(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & r \geq 0 \\ 0 & r < 0 \end{cases} \tag{3.172}$$

Example 3.24 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 3.39) obtained from Equation (3.162) as

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_x y(m)} \exp\left\{-\frac{(\ln y(m) - \mu_x)^2}{2\sigma_x^2}\right\} \tag{3.173}$$

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

$$\mu_x = \ln \mu_y - \frac{1}{2} \ln 1 + \sigma_y^2/\mu_y^2 \tag{3.174}$$

$$\sigma_x^2 = \ln\left(1 + \sigma_y^2/\mu_y^2\right) \tag{3.175}$$

$(\mu_x,\ \sigma_x^2)$, and $(\mu_y,\ \sigma_y^2)$ are the mean and variance of x and y respectively. The inverse mapping relations for the translation of means and variances of normal to log-normal variables are

$$\mu_y = \exp(\mu_x + \sigma_x^2/2) \tag{3.176}$$

$$\sigma_y^2 = \mu_x^2\,[\exp(\sigma_y^2) - 1] \tag{3.177}$$

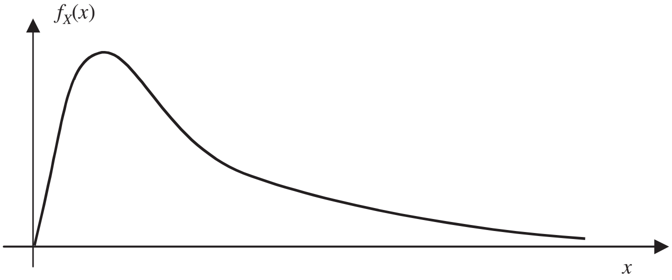


Figure 3.39 A log-normal probability density function.

3.8.2 *Many-to-One Mapping of Random Signals*

Now consider the case when the transform $h(\cdot)$ is a non-monotonic function such as that shown in Figure 3.40. 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

$$\text{Prob}\big(y(m) < Y < y(m) + \Delta y\big) = \sum_{k=1}^{K} \text{Prob}\big(x_k(m) < X < x_k(m) + \Delta x_k\big) \tag{3.178}$$

where x_k is the k^{th} root of $y(m) = h(x(m))$. Similar to the development in Section 3.7.1, Equation (3.178) can be written as

$$\frac{F_Y\big(y(m) + \Delta y\big) - F_Y\big(y(m)\big)}{\Delta y}\,\Delta y = \sum_{k=1}^{K} \frac{F_X\big(x_k(m) + \Delta x_k\big) - F_X\big(x_k(m)\big)}{\Delta x_k}\,\Delta x_k \tag{3.179}$$

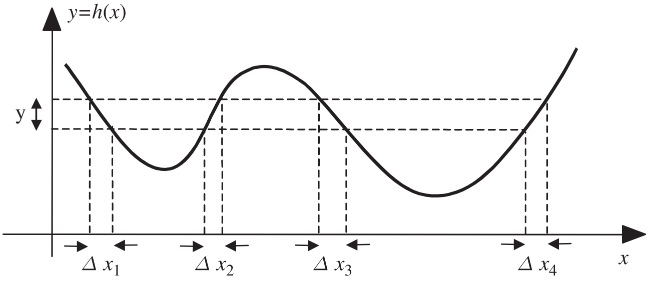


Figure 3.40 Illustration of a many-to-one transformation.

Equation (3.179) can be rearranged as

$$\frac{F_Y\big(y(m)+\Delta y\big)-F_Y\big(y(m)\big)}{\Delta y}=\sum_{k=1}^{K}\frac{\Delta x_k}{\Delta y}\,\frac{F_X\big(x_k(m)+\Delta x_k\big)-F_X\big(x_k(m)\big)}{\Delta x_k} \tag{3.180}$$

Now as the intervals Δx and Δy tend to zero Equation (3.180) becomes

$$\begin{aligned} f_Y\,(y(m)) &= \sum_{k=1}^{K}\left|\frac{\partial x_k(m)}{\partial y(m)}\right| f_X\big(x_k(m)\big) \\ &= \sum_{k=1}^{K}\frac{1}{|h'(x_k(m))|} f_X\big(x_k(m)\big) \end{aligned} \tag{3.181}$$

where $h'(x_k\,(m)) = \partial h(x_k(m))/\partial x_k(m)$. Note that Equation (3.179) is a generalised form of Equation (3.162); for a monotonic function, $K=1$ and Equation (3.179) becomes the same as Equation (3.162). Equation (3.181) can be expressed as

$$f_Y\big(y(m)\big)=\sum_{k=1}^{K}\left|J\big(x_k(m)\big)\right|^{-1} f_X\big(x_k(m)\big) \tag{3.182}$$

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

$$\boldsymbol{y}(m)=\boldsymbol{H}\big(\boldsymbol{x}(m)\big) \tag{3.183}$$

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

$$f_Y\big(\boldsymbol{y}(m)\big)=\sum_{k=1}^{K}\left|\boldsymbol{J}\big(\boldsymbol{x}_k(m)\big)\right|^{-1} f_X\big(\boldsymbol{x}_k(m)\big) \tag{3.184}$$

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

$$|\boldsymbol{J}\ (\boldsymbol{x})\,| = \begin{vmatrix} \dfrac{\partial y_1}{\partial x_1} & \dfrac{\partial y_1}{\partial x_2} & \cdots & \dfrac{\partial y_1}{\partial x_P} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial y_P}{\partial x_1} & \dfrac{\partial y_P}{\partial x_2} & \cdots & \dfrac{\partial y_P}{\partial x_P} \end{vmatrix} \tag{3.185}$$

For a monotonic linear vector transformation such as

$$\boldsymbol{y} = \boldsymbol{Hx} \tag{3.186}$$

the pdf of $\boldsymbol{y}$ becomes

$$f_Y(\boldsymbol{y}) = |\boldsymbol{J}(\boldsymbol{x})|^{-1} f_x\left(\boldsymbol{H}^{-1}\boldsymbol{y}\right) = |\boldsymbol{H}|^{-1} f_X(\boldsymbol{H}^{-1}\boldsymbol{y}) \tag{3.187}$$

where $|\boldsymbol{J(x)}| = \boldsymbol{H}$ is the Jacobian of the transformation and $|\boldsymbol{H}| = |\det(\boldsymbol{H})|$ where $\det(\boldsymbol{H})$ is the abbreviation of determinant of a matrix.

Example 3.25

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

$$\boldsymbol{y} = \boldsymbol{H}\,\boldsymbol{x} \tag{3.188}$$

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

$$f_X(\boldsymbol{x}) = \frac{1}{(2\pi)^{P/2}\,|\boldsymbol{\Sigma}_{xx}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x}\right) \tag{3.189}$$

From Equations (3.162) and (3.188)–(3.189), the pdf of the output $\boldsymbol{y}$ is given by

$$\begin{aligned} f_Y(\boldsymbol{y}) &= \frac{1}{(2\pi)^{P/2}\,|\boldsymbol{\Sigma}_{xx}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}^{-1^T}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{H}^{-1}\boldsymbol{y}\right)|\boldsymbol{H}|^{-1} \\ &= \frac{1}{(2\pi)^{P/2}\,|\boldsymbol{\Sigma}_{xx}|^{1/2}\,|\boldsymbol{H}|} \exp\left(-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{y}\right) \end{aligned} \tag{3.190}$$

where $\boldsymbol{\Sigma}_{yy} = \boldsymbol{H}\boldsymbol{\Sigma}_{xx}\boldsymbol{H}^{\mathrm{T}}$. Note that a linear transformation of a Gaussian process yields another scaled Gaussian process.

3.9 Search Engines: Citation Ranking

A search engine is a system designed to retrieve documents and files stored on computer systems such as on the World Wide Web, or a corporate computer network, or in a personal computer. The search engine allows the user to specify a set of keywords for searching a *list of items* (indexes) for contents in the database that match the keywords. This list of items is often sorted with respect to some measure of relevance of the results. Search engines use regularly updated indexes to operate efficiently.

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 help to focus the search on the documents and websites that contain the input keywords. However, the problem remains that often the contents of many websites are not of the required quality and furthermore there are misleading websites containing popular keywords aimed at attracting visitors in order to increase the hit rates and the advertising revenue of the sites.

For efficient information management the websites and their information content need to be ranked using an objective measure of quality. A well-established objective measure of quality of published information on 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 for the rapid calculation of a web page's rank in terms of citation. The page rank measure based on citation is a good way to organise the order of presentation of the results of an internet search.

3.9.1 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, so that the rank of a page represents a measure of the quality of the 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 on the web page is also an indicator.
(3) The number of citations of the web page on 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 is measured by its own citation by other web pages. Hence a citation from a much-cited source carries a greater weight.

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 within the tradition of academic research, a citation itself needs to be weighted by the quality of the citation's source i.e. by the citation's ranking of the source itself. A weighted citation gives some approximation of a page's importance or quality, where each source of a 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 $T_1, \ldots, T_n$ pointing to it. Page rank of A can be defined as

$$PR(A) = (1-d) + d\big(PR(T_1)/C(T_1) + \ldots + PR(T_n)/C(T_n)\big)$$

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 one. Page rank *PR(A)* can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalised link matrix of the web.

3.10 Summary

The theories of stochastic processes and statistical modelling are central to the development of signal processing methods and communication systems. We began this chapter with the definitions of some important concepts such as probability, information, deterministic signals, random signals and random processes. Probabilistic models and statistical measures, originally developed for random variables, were introduced and extended to model random signals.

Probability models were used to quantify information in terms of the entropy of a random process. The use of entropy in modelling information and in variable length coding was explained and several examples were presented.

We considered the concepts of stationary, ergodic-stationary and non-stationary processes. The concept of a stationary process is central to the theory of linear time-invariant systems and, furthermore, even non-stationary processes can be modelled by a chain of stationary sub-processes, as described in Chapter 11 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 employed extensively 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.

Box G.E.P and Jenkins G.M. (1976) *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.

Breiphol A.M. (1970) *Probabilistic System Analysis*. John Wiley & Sons, Inc, 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. John Wiley & Sons, Inc, New York.

Cooper G.R. and Mcgillem 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*. John Wiley & Sons, Inc, New York.

Jeffrey H. (1961) *Scientific Inference*, 3rd edn, Cambridge University Press.

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.

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**: 153–166.

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.

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*. John Wiley & Sons, Inc, New York.

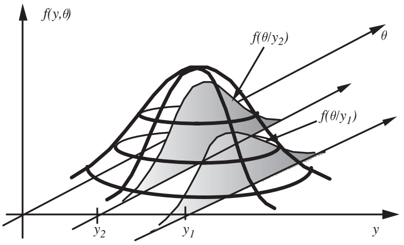
Rozanov Y.A. (1969) *Probability Theory: A Concise Course*. Dover, New York.

Shanmugan K.S. and Breipohl A.M. (1988) *Random Signals: Detection, Estimation and Data Analysis*. John Wiley & Sons, Inc, 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*. John Wiley & Sons, Inc, New York.

4



Bayesian Inference

Inference is the process of drawing conclusions from evidence. In signal processing the evidence is a function of the observation signal recorded by a number of sensors and the conclusion sought is the unknown values of some signals or parameters contained in the observation. Since the observations are usually noisy or incomplete, the conclusions inferred from the observations are subject to a level of uncertainty and error that depends on the quality of the observations and the efficiency of the inference method.

Bayesian inference is centred on Bayes' theorem derived from the works of Reverend Thomas Bayes, an 18th-century Presbyterian minister and mathematician. Bayes' solution to the problem of 'inverse probability' (this term is better known as posterior probability) was presented in his paper 'An Essay Towards Solving a Problem in the Doctrine of Chances' (1764), communicated posthumously by his friend Richard Price in a letter to John Canton for publication in Philosophical Transactions of the Royal Society of London, the letter begins with: 'Dear Sir, I now send you an essay which I have found among the papers of our deceased friend Mr. Bayes, and which, in my opinion, has great merit, and well deserves to be preserved…'.

Bayesian inference provides the general framework for formulation of probabilistic inference for prediction, estimation or classification of signals, or parameters from a set of observations. The Bayesian philosophy, for the prediction or estimation of a random process from a related signal, is based on combining the evidence contained in the observation signal and its likelihood function, with the prior knowledge of the probability distribution of the process and a cost of error function.

A classical example of Bayesian inference is the determination of the probability that a person has a disease given the evidence of a positive medical test. The probability of the positive test needs to be weighted by such priors as the probability of the disease in general or the probability of the disease for the particular medical group the person belongs to. The result should then be normalised by the probability of a positive test as

$$\text{Prob(disease|positive test)} = \text{Prob(positive test|disease)} \times \text{Prob(disease)} \big/ \text{Prob(positive test)}$$

It can be argued that Bayesian inference resembles the human reasoning process since people often weight the evidence of an observation with their accumulated prior beliefs or dispositions. Sometimes a strong prior can admit weak evidence that happens to be in agreement with the prior or reject strong evidence that disagrees with the prior. Indeed, the form of the prior functions used in a Bayesian estimation

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

can be an objective description of the distribution of the variables or it can be a subjective description based on a person's belief, experience or intuitive feeling.

Bayesian inference is based on minimisation of the Bayes' risk function, which includes a probability model of the unknown parameters conditional on the given observation and a cost-of-error function. As illustrated in Figure 4.1, 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 its special cases. The hidden Markov model, widely used in pattern recognition, is an example of a Bayesian model.

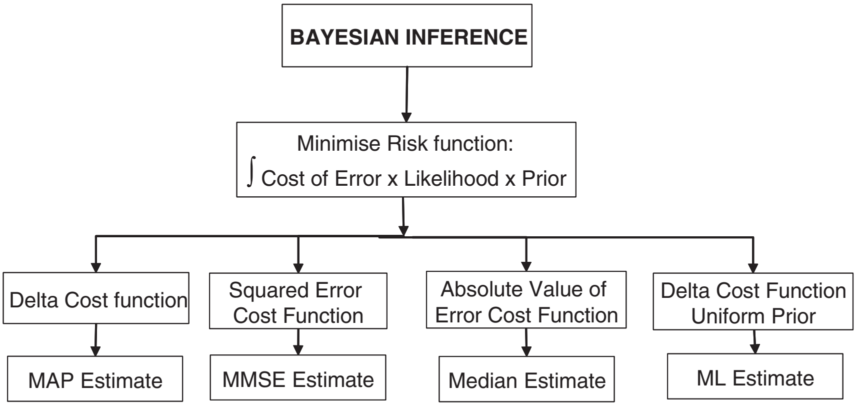

BAYESIAN INFERENCE
Minimise Risk function:
∫ Cost of Error x Likelihood x Prior
Delta Cost function
Squared Error Cost Function
Absolute Value of Error Cost Function
Delta Cost Function Uniform Prior
MAP Estimate
MMSE Estimate
Median Estimate
ML Estimate

Figure 4.1 Bayesian inference involves a cost function, a prior function and a likelihood function. As illustrated, other probabilistic estimation methods may be considered as special cases of Bayesian estimation.

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 estimation of the mixture Gaussian model 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 a related 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 or phase) as in radar signal processing applications, or we may wish to recover the signal itself.

Note that the best estimate is the one that minimises a risk function or a cost of error function and that often involves the use of probability of the unknown vector given the known observation, as explained in this chapter.

An estimator takes as input a set of noisy or incomplete observations, and, using a dynamic model of the signal (e.g. a linear predictive model) and/or a probabilistic model of the distribution of the process (e.g. Gaussian model), estimates the unknown parameters. The estimation accuracy depends on the available information, the accuracy of the models and the efficiency of the estimator. In this chapter, the

Bayesian estimation of stationary parameters is studied. The modelling and estimation of non-stationary finite-state processes is covered in Chapter 5 on hidden Markov models (HMMs).

Bayesian theory is a general inference framework for the derivation of statistical estimation methods, as illustrated in Figure 4.1. The various forms of statistical estimators can be derived as special cases of Bayesian estimation.

4.1.1 Bayes' Theorem

Bayes' theorem, also known as Bayes' rule, is based on the works of Thomas Bayes. Consider the estimation of the value of a random parameter vector $\boldsymbol{\theta}$, given a related observation vector $\boldsymbol{y}$. From Bayes' rule (introduced in Chapter 3) the posterior probability density function (pdf) of the parameter vector $\boldsymbol{\theta}$ given $\boldsymbol{y}$, $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$, can be expressed as

$$\underbrace{f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})}_{Posterior} = \frac{1}{f_Y(\boldsymbol{y})}\underbrace{f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}_{Likelihood}\underbrace{f_{\Theta}(\boldsymbol{\theta})}_{Prior} \tag{4.1}$$

where for a given observation, $f_Y(\boldsymbol{y}) = \int f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})d\boldsymbol{\theta}$ is a constant and has only a normalising effect. Thus there are two variable terms in Equation (4.1): one term $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood that the observation signal $\boldsymbol{y}$ 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 influences of the likelihood pdf $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior pdf $f_{\Theta}(\boldsymbol{\theta})$ on the posterior pdf $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$ depends on the shape of these functions, i.e. on how relatively peaked each pdf is. In general the more peaked a probability density function, the more it will influence the outcome of the estimation process. Conversely, a uniform prior pdf will have no influence other than rejecting those values of the parameters which fall outside the pdf and have zero. The roles of the posterior, prior and likelihood functions are explained in more detail in Section 4.1.5.

4.1.2 Elements of Bayesian Inference

The Bayesian inference can be derived from the minimisation of a risk function defined as the cost of error averaged over all values of the unknown parameter vector as

$$Risk(\hat{\boldsymbol{\theta}}) = \int_{\theta}\int_{Y} Cost(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\underbrace{f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}_{Likelihood}\underbrace{f_{\Theta}(\boldsymbol{\theta})}_{Prior}\ d\boldsymbol{y}\ d\boldsymbol{\theta} \tag{4.2}$$

The elements of Bayesian inference are as follows:

Bayesian Risk: The risk in making an estimate $\hat{\boldsymbol{\theta}}$ of the unknown value of the parameter vector $\boldsymbol{\theta}$, is related to the probability of error and the cost of error. Since the true value of the parameters $\boldsymbol{\theta}$ is unknown, the cost needs to be averaged over all possible values of $\boldsymbol{\theta}$ as shown in Equation (4.2).

Cost function: The cost of error determines the type of Bayesian solution. A popular cost function is the minimum mean squared error cost. The cost function can be chosen to give very high costs to unacceptable errors with catastrophic consequences.

Likelihood: The likelihood, $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$, is the conditional probability of the observation $\boldsymbol{y}$ given the parameter $\boldsymbol{\theta}$. In other words, it gives the likelihood that the observation signal $\boldsymbol{y}$ is due to parameter $\boldsymbol{\theta}$. The likelihood does not take into account then probability of $\boldsymbol{\theta}$, $f_{\Theta}(\boldsymbol{\theta})$, known as its prior.

Prior probability: The prior function, $f_{\Theta}(\boldsymbol{\theta})$, gives the probability density function of the parameter vector $\boldsymbol{\theta}$. The prior acts as a 'moderating' influence on the likelihood function. It will be shown later that the extent to which the likelihood and the prior functions influence the estimate depends on such factors as the signal-to-noise ratio, the length of the observation and the shape of the prior.

Posterior probability: The posterior probability, $f_{Y|\Theta}(\boldsymbol{\theta}|\boldsymbol{y}) = f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})/f_Y(\boldsymbol{y})$, is proportional to the product of the likelihood and the prior as expressed by Bayes' rule in Equation (4.1).

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.

4.1.3 Dynamic and Probability Models in Estimation

Optimal estimation algorithms, such as Kalman filters, utilise both 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 sinusoidal models and linear prediction models.

A statistical probability model characterises the space of different realisations of a random 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 the values of some other related parameters or 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, introduced in Chapter 5) of the probability distribution of the signal and sinusoidal or linear prediction models of the dynamics of the signal. A further example is a Kalman filter (Chapter 7) that utilises state-space equations for the dynamics of the signal and noise and assumes that the random processes have Gaussian distributions.

As an illustration consider the estimation of a P-dimensional parameter vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{P-1}]$ from a noisy observation vector $\boldsymbol{y} = [y(0), y(1), \ldots, y(N-1)]$ modelled as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n} = h(\boldsymbol{\theta}, \boldsymbol{e}) + \boldsymbol{n} \tag{4.3}$$

where, as illustrated in Figure 4.2, it is assumed that the clean signal $\boldsymbol{x}$ is the output of a predictive model $h(\cdot)$ with a random input $\boldsymbol{e}$ and parameter vector $\boldsymbol{\theta}$ and $\boldsymbol{n}$ is an additive random noise process. In Figure 4.2, the distributions of the random input $\boldsymbol{e}$, the parameter vector $\boldsymbol{\theta}$ and the random noise $\boldsymbol{n}$ are

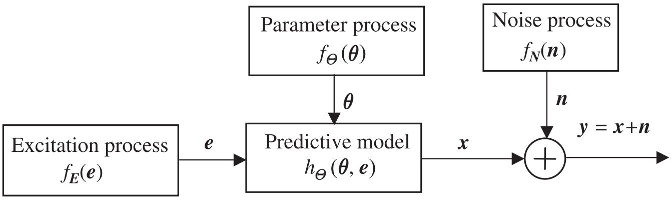


Figure 4.2 A random process $\boldsymbol{y}$ is described in terms of a predictive model $h(\cdot)$, of signal trajectory and statistical models of the random fluctuations $f_E(\cdot)$, $f_{\Theta}(\cdot)$ and $f_N(\cdot)$.

modelled by probability density functions $f_E(\cdot)$, $f_\Theta(\cdot)$ and $f_N(\cdot)$ respectively. The pdf model most often used is the Gaussian model.

Predictive and statistical models of a process guide the estimator towards the set of values of the unknown parameters that are most consistent with both the prior distribution of the model parameters and the observation. In general, the more modelling information employed 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 then more harm than good may result from using them.

4.1.4 Parameter Space and Signal Space

Consider a random process with a parameter vector $\boldsymbol{\theta}$. For example, each instance of $\boldsymbol{\theta}$ could be the parameter vector for a dynamic model, such as a harmonic model, of a speech sound or a musical note. The parameter space of a process $\boldsymbol{\Theta}$ is the collection of all the values that the parameter vector $\boldsymbol{\theta}$ 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 $\boldsymbol{\theta}$ of a process has an associated signal space $\boldsymbol{Y}$; this is the collection of all the signal realisations of the process with the parameter value $\boldsymbol{\theta}$.

For example, consider a three-dimensional vector-valued Gaussian process with parameter vector $\boldsymbol{\theta} = [\boldsymbol{\mu}, \ \boldsymbol{\Sigma}]$, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix of the Gaussian process. Figure 4.3 illustrates three mean vectors in a three-dimensional 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.

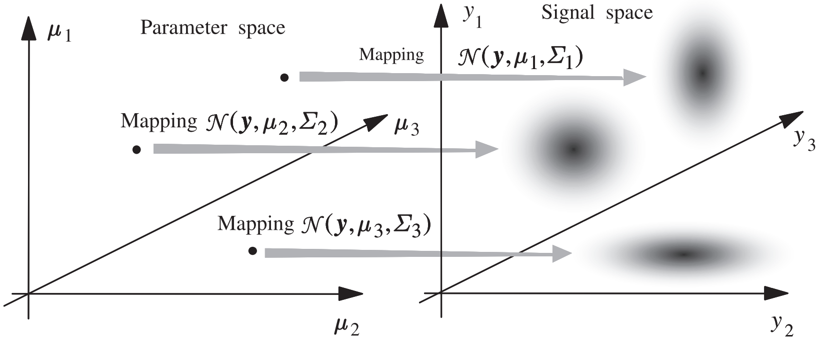

μ_1
Parameter space
y_1
Signal space
Mapping
$\mathcal{N}(\boldsymbol{y}, \mu_1, \Sigma_1)$
Mapping $\mathcal{N}(\boldsymbol{y}, \mu_2, \Sigma_2)$
μ_3
y_3
Mapping $\mathcal{N}(\boldsymbol{y}, \mu_3, \Sigma_3)$
μ_2
y_2

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.

4.1.5 Parameter Estimation and Signal Restoration

Parameter estimation and signal restoration are closely related problems. The main differences arise 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 known samples, as illustrated in Figure 4.4. Assume that an autoregressive (AR) process is used to model the signal as

$$\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{e} + \boldsymbol{n} \tag{4.4}$$

where $\boldsymbol{y}$ is the observation signal vector, $\boldsymbol{X}$ is the clean signal matrix, $\boldsymbol{\theta}$ is the AR parameter vector, $\boldsymbol{e}$ is the random input of the AR model and $\boldsymbol{n}$ 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 $\boldsymbol{e}$ 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 $\boldsymbol{e}$ of the signal $\boldsymbol{x}$ is purely random and, unlike $\boldsymbol{\theta}$, it cannot be estimated through an averaging operation.

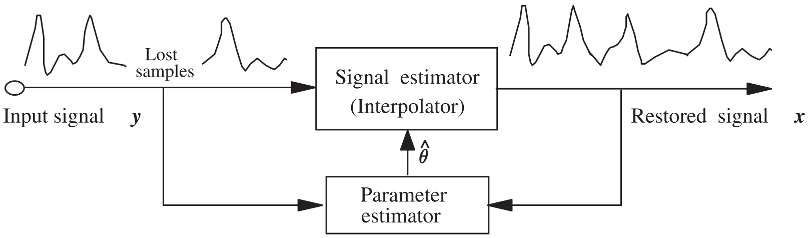

Lost samples
Input signal $\boldsymbol{y}$
Signal estimator (Interpolator)
$\hat{\theta}$
Parameter estimator
Restored signal $\boldsymbol{x}$

Figure 4.4 Illustration of signal restoration using a parametric model of the signal.

4.1.6 Performance Measures and Desirable Properties of Estimators

In estimation of a parameter vector $\boldsymbol{\theta}$ from N observation samples $\boldsymbol{y}$, 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 $\boldsymbol{y}$, the length of the observation N and the process model $\mathcal{M}$. This dependence may be expressed as

$$\hat{\boldsymbol{\theta}} = f(\boldsymbol{y}, N, \mathcal{M}) \tag{4.5}$$

Different parameter estimators produce different results depending on the estimation method and utilisation of the observation and the influence of the prior information. Due to 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 probability density function. 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: $\mathcal{E}[\hat{\boldsymbol{\theta}}]$
(2) Bias of estimate: $\mathcal{E}[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] = \mathcal{E}[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta}$
(3) Covariance of estimate: $\text{Cov}\,[\hat{\boldsymbol{\theta}}] = \mathcal{E}[(\hat{\boldsymbol{\theta}} - \mathcal{E}[\hat{\boldsymbol{\theta}}])(\hat{\boldsymbol{\theta}} - \mathcal{E}[\hat{\boldsymbol{\theta}}])^{\mathrm{T}}]$

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 $\boldsymbol{\theta}$ is unbiased if the expectation of the estimate is equal to the true parameter value:

$$\mathcal{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta} \tag{4.6}$$

An estimator is asymptotically unbiased if for increasing length of observations N we have

$$\lim_{N\to\infty} \mathcal{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta} \tag{4.7}$$

(2) Efficient estimator: an unbiased estimator of $\boldsymbol{\theta}$ is an efficient estimator if it has the smallest covariance matrix compared with all other unbiased estimates of $\hat{\boldsymbol{\theta}}$:

$$\text{Cov}[\hat{\boldsymbol{\theta}}_{\text{Efficient}}] \le \text{Cov}[\hat{\boldsymbol{\theta}}] \tag{4.8}$$

where $\hat{\boldsymbol{\theta}}$ is any other estimate of $\boldsymbol{\theta}$.
(3) Consistent estimator: an estimator is consistent if the estimate improves with the increasing length of the observation N, such that the estimate $\hat{\boldsymbol{\theta}}$ converges probabilistically to the true value $\boldsymbol{\theta}$ as N becomes infinitely large:

$$\lim_{N\to\infty} P[|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}| > \varepsilon] = 0 \tag{4.9}$$

where ε is arbitrary small.

Example 4.1 Estimation of the mean and variance of a signal

Consider the bias in the time-averaged estimates of the mean μ_y and the variance σ_y^2 of N observation samples $[y(0), \ldots, y(N-1)]$, of an ergodic random process, given as

$$\hat{\mu}_y = \frac{1}{N}\sum_{m=0}^{N-1} y(m) \tag{4.10}$$

$$\hat{\sigma}_y^2 = \frac{1}{N}\sum_{m=0}^{N-1} \left[y(m) - \hat{\mu}_y\right]^2 \tag{4.11}$$

It is easy to show that $\hat{\mu}_y$ is an unbiased estimate, since

$$\mathcal{E}\left[\hat{\mu}_y\right] = \frac{1}{N}\sum_{m=0}^{N-1} \mathcal{E}[y(m)] = \mu_y \tag{4.12}$$

The expectation of the estimate of the variance $\hat{\sigma}_y^2$ can be expressed as

$$\begin{aligned}\mathcal{E}\left[\hat{\sigma}_y^2\right] &= \mathcal{E}\left[\frac{1}{N}\sum_{m=0}^{N-1}\left(y(m)-\frac{1}{N}\sum_{k=0}^{N-1}y(k)\right)^2\right]\\ &= \sigma_y^2 - \frac{2}{N}\sigma_y^2 + \frac{1}{N}\sigma_y^2\\ &= \sigma_y^2 - \frac{1}{N}\sigma_y^2 \end{aligned} \tag{4.13}$$

From Equation (4.13), 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.

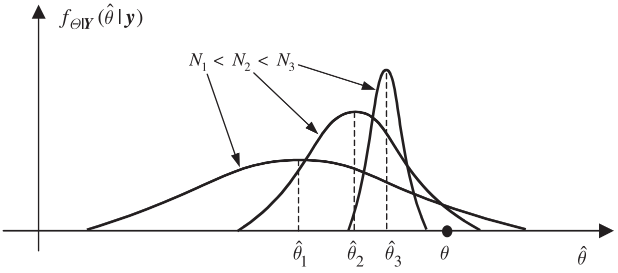


Figure 4.5 Illustration of the decrease in the bias and variance of an asymptotically unbiased estimate of the parameter θ with increasing length of observation.

4.1.7 Prior and Posterior Spaces and Distributions

The Bayesian inference method weights the likelihood that a value of parameter $\boldsymbol{\theta}$ underlies an observation $\boldsymbol{y}$, $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$, with the prior probability of the value of the parameters $f_{\Theta}(\boldsymbol{\theta})$. 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 non-uniform distribution. If the prior distribution is non-uniform (e.g. Gaussian, Gamma, etc) then the non-uniform 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 an observation signal $\boldsymbol{x}$, or a parameter vector $\boldsymbol{\theta}$, is contained in the observation signal $\boldsymbol{y}$ 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 $\boldsymbol{y}$ may be used to obtain an estimate of the clean speech $\boldsymbol{x}$ and/or the parameter vector $\boldsymbol{\theta}$ 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 $\boldsymbol{x}$, or a parameter vector $\boldsymbol{\theta}$, that are consistent with both the prior information on signal $\boldsymbol{x}$ (or parameter $\boldsymbol{\theta}$)

and the evidence contained in the observation $\boldsymbol{y}$. The significance of posterior probability is that the likelihood $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ of each value of a parameter $\boldsymbol{\theta}$ is weighted with the prior probability of the value $f_{\Theta}(\boldsymbol{\theta})$.

For example, the likelihood that a variable, such as the state of tomorrow's weather $\boldsymbol{\theta}$, takes a particular value given some related meteorological observations $\boldsymbol{y}$, (i.e. probability of $\boldsymbol{\theta}$ given $\boldsymbol{y}$, $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$), can be obtained using the likelihood that meteorological observation $\boldsymbol{y}$ can be seen in the weather state $\boldsymbol{\theta}$ (i.e. $\boldsymbol{y}$ given $\boldsymbol{\theta}$, $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$) weighted by the prior likelihood of the weather state $\boldsymbol{\theta}$ (i.e. $f_{\Theta}(\boldsymbol{\theta})$) (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 $\boldsymbol{\Theta}$, observation space $\boldsymbol{Y}$ and a joint pdf $f_{Y,\theta}(\boldsymbol{y},\boldsymbol{\theta})$. From Bayes' rule the posterior pdf of the parameter vector $\boldsymbol{\theta}$, given an observation vector $\boldsymbol{y}$, $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$, can be expressed as

$$f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})f_{\Theta}(\boldsymbol{\theta})}{f_Y(\boldsymbol{y})} \tag{4.14}$$

where, for a given observation vector $\boldsymbol{y}$, the pdf $f_Y(\boldsymbol{y})$ is a constant and has only a normalising effect. From Equation (4.14), the posterior pdf is proportional to the weighted likelihood; that is the product of the likelihood $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ that the observation $\boldsymbol{y}$ was generated by the parameter vector $\boldsymbol{\theta}$, and the prior pdf $f_{\Theta}(\boldsymbol{\theta})$. The prior pdf gives the unconditional parameter distribution averaged over the entire observation space as

$$f_{\Theta}(\boldsymbol{\theta}) = \int_{Y} f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})d\boldsymbol{y} \tag{4.15}$$

For most applications, it is relatively convenient to obtain the likelihood function $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$. The prior pdf influences the inference drawn from the likelihood function by weighting it with $f_{\Theta}(\boldsymbol{\theta})$. 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 $\boldsymbol{y}$ cuts a posterior pdf $f_{\Theta|Y}(\theta|\boldsymbol{y})$ through the joint distribution.

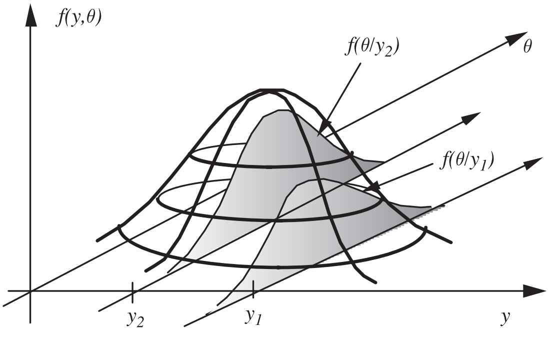


Figure 4.6 Illustration of joint distribution of a signal y and a parameter θ and the posterior distribution of θ given y.

Example 4.2

A noisy signal vector of length N samples is modelled as

$$\boldsymbol{y}(m) = \boldsymbol{x}(m) + \boldsymbol{n}(m) \tag{4.16}$$

Assume that the signal $\boldsymbol{x}(m)$ is Gaussian with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the noise $\boldsymbol{n}(m)$ 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 $\boldsymbol{y}(m)$, the underlying signal $\boldsymbol{x}(m)$ would have a likelihood distribution with a mean vector of $\boldsymbol{y}(m) - \boldsymbol{\mu}_n$ and a covariance matrix $\boldsymbol{\Sigma}_{nn}$ as shown in Figure 4.7. The likelihood function is given by

$$\begin{aligned} f_{Y|X}\big(\boldsymbol{y}(m)\,|\boldsymbol{x}(m)\big) &= f_N\big(\boldsymbol{y}(m) - \boldsymbol{x}(m)\big) \\ &= \frac{1}{(2\pi)^{N/2}\,|\boldsymbol{\Sigma}_{nn}|^{1/2}} \exp\left\{-\frac{1}{2}\left[\big(\boldsymbol{y}(m) - \boldsymbol{\mu}_n\big) - \boldsymbol{x}(m)\right]^{\mathrm{T}} \boldsymbol{\Sigma}_{nn}^{-1} \left[\big(\boldsymbol{y}(m) - \boldsymbol{\mu}_n\big) - \boldsymbol{x}(m)\right]\right\} \end{aligned} \tag{4.17}$$

where the terms in the exponential function have been rearranged to emphasize the illustration of the likelihood space in Figure 4.4. Hence the posterior pdf can be expressed as

$$\begin{aligned} \underbrace{f_{X|Y}\big(\boldsymbol{x}(m)|\boldsymbol{y}(m)\big)}_{\text{Posterior}} &= \frac{1}{f_Y\big(\boldsymbol{y}(m)\big)} \underbrace{f_{Y|X}\big(\boldsymbol{y}(m)|\boldsymbol{x}(m)\big)}_{\text{Likeihood}} \underbrace{f_X\big(\boldsymbol{x}(m)\big)}_{\text{Prior}} \\ &= \frac{1}{f_Y\big(\boldsymbol{y}(m)\big)} \frac{1}{(2\pi)^N |\boldsymbol{\Sigma}_{nn}|^{1/2} |\boldsymbol{\Sigma}_{xx}|^{1/2}} \\ &\quad \times \exp\left(-\frac{1}{2}\left\{\underbrace{\left[\big(\boldsymbol{y}(m) - \boldsymbol{\mu}_n\big) - \boldsymbol{x}(m)\right]^T \boldsymbol{\Sigma}_{nn}^{-1}\left[\big(\boldsymbol{y}(m) - \boldsymbol{\mu}_n\big) - \boldsymbol{x}(m)\right]}_{\text{Likelihood}} \right.\right. \\ &\quad \left.\left. + \underbrace{\big(\boldsymbol{x}(m) - \boldsymbol{\mu}_x\big)^T \boldsymbol{\Sigma}_{xx}^{-1}\big(\boldsymbol{x}(m) - \boldsymbol{\mu}_x\big)}_{\text{Prior}}\right\}\right) \end{aligned} \tag{4.18}$$

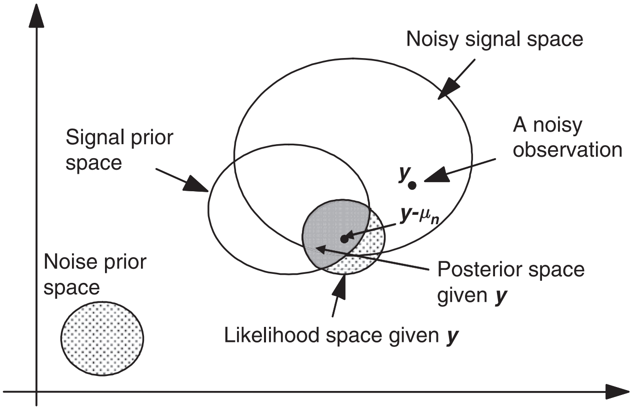


Figure 4.7 Sketch of two-dimensional signal and noise prior spaces and the likelihood and posterior spaces of a noisy observation $\boldsymbol{y}$.

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. Note in this figure that the mean (i.e. centre) and the variance (related to the size of the ellipse) of the noisy signal space are the sum of the means and variances of the signal and noise spaces respectively. The likelihood and the posterior spaces for a noisy observation vector $\boldsymbol{y}$ are also illustrated in Figure 4.7. The likelihood space, i.e. the space of the likely values of the signal, is centred at $\boldsymbol{y}(m) - \boldsymbol{\mu}_n$ and has a size proportional to the variance of the noise $\boldsymbol{\Sigma}_{nn}$. The clean signal is then somewhere within the posterior subspace determined by the intersection of the likelihood and the prior space of the signal.

4.2 Bayesian Estimation

The Bayesian estimation of a parameter vector $\boldsymbol{\theta}$, from a related observation vector $\boldsymbol{y}$, is based on minimisation of a Bayesian risk function defined as an average cost-of-error function:

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{\theta}}) &= \mathcal{E}[C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})] \\ &= \int_{\theta}\int_{Y} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{Y,\Theta}(\boldsymbol{y}, \boldsymbol{\theta})\ d\boldsymbol{y}\ d\boldsymbol{\theta} \\ &= \int_{\theta}\int_{Y} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta})\ d\boldsymbol{y}\ d\boldsymbol{\theta}\end{aligned} \tag{4.19}$$

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 of the estimator. The cost function can be chosen to associate higher costs with those outcomes that are undesirable or disastrous, such as a high cost for a false diagnosis in a medical test. In Equation (4.19) the Bayes' risk function is averaged over the space of all values of the parameter $\boldsymbol{\theta}$ and the observation $\boldsymbol{y}$.

For a given $\boldsymbol{y}$, $f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})$ is a constant and has no effect on the risk-minimisation process. Hence substituting $f_{Y,\Theta}(\boldsymbol{y}, \boldsymbol{\theta}) = f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y}) f_Y(\boldsymbol{y})$ in the (middle) line of Equation (4.19) a conditional risk function is obtained as

$$\mathcal{R}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) = \int_{\theta} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\ d\boldsymbol{\theta} \tag{4.20}$$

The Bayesian estimate $\hat{\boldsymbol{\theta}}$ is obtained as the minimum-risk parameter vector given by

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\theta}}\ \mathcal{R}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) = \arg\min_{\hat{\theta}} \left[\int_{\theta} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\ d\boldsymbol{\theta}\right] \tag{4.21}$$

Using Bayes' rule, Equation (4.21) can be written as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\theta}} \left[\int_{\theta} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta})\ d\boldsymbol{\theta}\right) \tag{4.22}$$

Assuming that the risk function is differentiable, and has a well-defined minimum, the Bayesian estimate can be obtained as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\operatorname*{zero}_{\hat{\theta}} \frac{\partial \mathcal{R}(\hat{\boldsymbol{\theta}}|\boldsymbol{y})}{\partial \hat{\boldsymbol{\theta}}} = \arg\operatorname*{zero}_{\hat{\theta}} \left[\frac{\partial}{\partial \hat{\boldsymbol{\theta}}} \int_{\theta} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\Theta}(\boldsymbol{\theta})\ d\boldsymbol{\theta}\right] \tag{4.23}$$

4.2.1 Maximum A Posteriori Estimation

The maximum a posteriori (MAP) estimate $\hat{\boldsymbol{\theta}}_{MAP}$ is obtained as the parameter vector that maximises the posterior pdf. 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

$$C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = 1 - \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \tag{4.24}$$

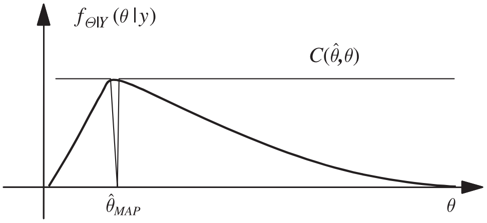


Figure 4.8 Illustration of the Bayesian cost function for the MAP estimate.

where $\delta(\hat{\boldsymbol{\theta}},\ \boldsymbol{\theta})$ is the Kronecker delta function. Substitution of the cost function in the Bayesian risk equation yields

$$\begin{aligned}\mathcal{R}_{\mathrm{MAP}}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right) &= \int_{\boldsymbol{\theta}}[1-\delta(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})]f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\,d\boldsymbol{\theta} \\ &= 1-f_{\boldsymbol{\Theta}|Y}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right)\end{aligned} \tag{4.25}$$

From Equation (4.25), 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 (4.25) or equivalently maximisation of the posterior function

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} &= \arg\max_{\boldsymbol{\theta}} f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{\theta}}[f_{Y|\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{\theta})f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})]\end{aligned} \tag{4.26}$$

4.2.2 Maximum-Likelihood (ML) Estimation

The maximum-likelihood (ML) estimate $\hat{\boldsymbol{\theta}}_{\mathrm{ML}}$ is obtained as the parameter vector that maximises the likelihood function $f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$. The ML estimator corresponds to a Bayesian estimator with a notch-shaped cost function and a uniform parameter prior pdf:

$$\begin{aligned}\mathcal{R}_{\mathrm{ML}}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) &= \int_{\boldsymbol{\theta}} \underbrace{[1-\delta(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})]}_{\text{cost function}} \underbrace{f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}_{\text{Likelihood}} \underbrace{f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}_{\text{prior}}\,d\boldsymbol{\theta} \\ &= \text{const.}[1-f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\hat{\boldsymbol{\theta}})]\end{aligned} \tag{4.27}$$

where the prior function $f_{\Theta}(\boldsymbol{\theta}) = \text{const}$.

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 (4.27), it is evident that minimisation of the risk function is achieved by maximisation of the likelihood function

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \tag{4.28}$$

In practice it is convenient to maximise the log-likelihood function instead of the likelihood:

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \log\left(f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\right) \tag{4.29}$$

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.
(3) Unlike the likelihood function, the log-likelihood has a dynamic range that does not cause computational underflow.

Note that dynamic range is the range of minimum and maximum values. The multiplication of extremely small numbers can yield a result that is too small for computer numerical representation resulting in an underflow. Conversely the multiplication of very large numbers can yield a number that is too big for computer representation resulting in an overflow.

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 and the covariance matrix of a P-dimensional Gaussian vector process $\boldsymbol{y}(m)$ from N observation vectors $[\boldsymbol{y}(0), \boldsymbol{y}(1), \ldots, \boldsymbol{y}(N-1)]$. Assuming the observation vectors are uncorrelated, the pdf of the observation sequence is given by

$$f_Y\left(\boldsymbol{y}(0),\ \cdots,\boldsymbol{y}(N-1)\right)=\prod_{m=0}^{N-1}\frac{1}{(2\pi)^{P/2}\left|\boldsymbol{\Sigma}_{yy}\right|^{1/2}}\exp\left\{-\frac{1}{2}\left[\boldsymbol{y}(m)-\mu_y\right]^{\mathrm{T}}\boldsymbol{\Sigma}_{yy}^{-1}\left[\boldsymbol{y}(m)-\mu_y\right]\right\} \tag{4.30}$$

and the log-likelihood equation is given by

$$\ln f_Y(\boldsymbol{y}(0)\ldots,\boldsymbol{y}(N-1))=\sum_{m=0}^{N-1}\left\{-\frac{P}{2}\ln(2\pi)-\frac{1}{2}\ln\left|\boldsymbol{\Sigma}_{yy}\right|-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_y\right]^{T}\boldsymbol{\Sigma}_{yy}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_y\right]\right\} \tag{4.31}$$

Taking the derivative of the log-likelihood equation with respect to the mean vector yields

$$\frac{\partial\ln f_Y(\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1))}{\partial\boldsymbol{\mu}_y}=\sum_{m=0}^{N-1}\left[-2\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{y}(m)+2\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\mu}_y\right]=0 \tag{4.32}$$

From Equation (4.32), we have

$$\hat{\boldsymbol{\mu}}_y=\frac{1}{N}\sum_{m=0}^{N-1}\boldsymbol{y}(m) \tag{4.33}$$

To obtain the ML estimate of the covariance matrix we take the derivative of the log-likelihood equation with respect to

$$\frac{\partial\ \ln f_Y\left(\boldsymbol{y}(0),\cdots,\boldsymbol{y}(N-1)\right)}{\partial\ \boldsymbol{\Sigma}_{yy}^{-1}}=\sum_{m=0}^{N-1}\left\{\frac{1}{2}\boldsymbol{\Sigma}_{yy}-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_y\right]\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_y\right]^{\mathrm{T}}\right\}=0 \tag{4.34}$$

From Equation (4.34), we have an estimate of the covariance matrix as

$$\hat{\boldsymbol{\Sigma}}_{yy}=\frac{1}{N}\sum_{m=0}^{N-1}\left[\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_y\right]\left[\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_y\right]^{\mathrm{T}} \tag{4.35}$$

Example 4.4 ML and MAP estimation of a Gaussian random parameter

Consider the estimation of a P-dimensional random parameter vector $\boldsymbol{\theta}$ from an N-dimensional observation vector $\boldsymbol{y}$. Assume that the relation between the signal vector $\boldsymbol{y}$ and the parameter vector $\boldsymbol{\theta}$ is described by a linear model as

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{\theta} + \boldsymbol{e} \tag{4.36}$$

where $\boldsymbol{e}$ is a random excitation input signal and $\boldsymbol{G}$ is a data matrix. For example for an autoregressive process, $\boldsymbol{G}$ would be composed of past values of $\boldsymbol{y}$. The pdf of the parameter vector $\boldsymbol{\theta}$ given an observation vector $\boldsymbol{y}$ can be described, using Bayes' rule, as

$$f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{1}{f_Y(\boldsymbol{y})} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \tag{4.37}$$

Assuming that the matrix $\boldsymbol{G}$ in Equation (4.36) is known, the likelihood of the signal $\boldsymbol{y}$ given the parameter vector $\boldsymbol{\theta}$ is the pdf of the random vector:

$$f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) = f_E(\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta}) \tag{4.38}$$

Now assume the input signal vector $\boldsymbol{e}$ 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 of $\boldsymbol{\mu_\theta}$ and covariance matrix $\boldsymbol{\Sigma_{\theta\theta}}$. Therefore, the likelihood function equation (4.38) can be written as

$$f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) = f_E(\boldsymbol{e}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left[-\frac{1}{2\sigma_e^2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta})\right] \tag{4.39}$$

and

$$f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^T \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right] \tag{4.40}$$

The ML estimate, obtained from maximisation of the log-likelihood function $\ln\left[f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\right]$ with respect to $\boldsymbol{\theta}$, is given by

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}}(\boldsymbol{y}) = (\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} \tag{4.41}$$

To obtain the MAP estimate we first form the posterior probability distribution by substituting Equations (4.39) and (4.40) in Equation (4.37):

$$\begin{aligned} f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y}) = {} & \frac{1}{f_Y(\boldsymbol{y})} \frac{1}{(2\pi\sigma_e^2)^{N/2}} \frac{1}{(2\pi)^{P/2}\left|\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}\right|^{1/2}} \\ & \times \exp\left(-\frac{1}{2\sigma_e^2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})\right) \end{aligned} \tag{4.42}$$

The MAP parameter estimate, obtained by differentiating the log-likelihood function $\ln f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})$ and setting the derivative to zero, is given by

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}(\boldsymbol{y}) = (\boldsymbol{G}^T\boldsymbol{G} + \sigma_e^2\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1})^{-1}\left(\boldsymbol{G}^T\boldsymbol{y} + \sigma_e^2\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\theta}}\right) \tag{4.43}$$

Note that as the covariance of the Gaussian-distributed parameter increases, or equivalently as $\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1} \to 0$, the Gaussian prior tends to a uniform prior and the MAP solution Equation (4.43) tends to the ML solution given by Equation (4.41). Conversely as the pdf of the parameter vector $\boldsymbol{\theta}$ becomes more peaked, i.e. as $\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} \to 0$, the estimate tends towards $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ the mean of the prior pdf.

4.2.3 Minimum Mean Square Error Estimation

The Bayesian minimum mean square error (MMSE) estimate is obtained as the parameter vector that minimises a mean square error cost function (Figure 4.9) defined as

$$\begin{aligned}\mathcal{R}_{\text{MSE}}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) &= \mathcal{E}[(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})^2|\boldsymbol{y}]\\ &= \int_{\boldsymbol{\theta}} (\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})^2 f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}\end{aligned} \tag{4.44}$$

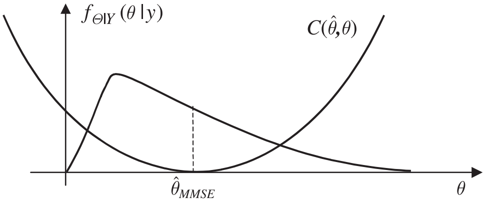


Figure 4.9 Illustration of the mean square error cost function and estimate.

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:

$$\frac{\partial\,\mathcal{R}_{\text{MSE}}(\hat{\boldsymbol{\theta}}|\boldsymbol{y})}{\partial\hat{\boldsymbol{\theta}}} = 2\hat{\boldsymbol{\theta}} \underbrace{\int_{\boldsymbol{\theta}} f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}}_{1} = 2\int_{\boldsymbol{\theta}} \boldsymbol{\theta} f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.45}$$

Since the first integral on the right-hand side of Equation (4.45) is equal to 1, we have

$$\frac{\partial R_{\text{MSE}}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right)}{\partial\hat{\boldsymbol{\theta}}} = 2\hat{\boldsymbol{\theta}} - 2\int_{\boldsymbol{\theta}} \boldsymbol{\theta} f_{\boldsymbol{\Theta}|Y}(d\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.46}$$

The MMSE solution is obtained by setting Equation (4.46) to zero:

$$\hat{\boldsymbol{\theta}}_{\text{MMSE}}(\boldsymbol{y}) = \int_{\boldsymbol{\theta}} \boldsymbol{\theta}\, f_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.47}$$

For cases where we do not have the pdf models of the parameter process $\boldsymbol{\theta}$ and the signal $\boldsymbol{y}$, the minimum mean square error (known as the least square error, LSE) estimate is obtained through minimisation of a mean square error function:

$$\hat{\boldsymbol{\theta}}_{\text{LSE}} = \arg\min_{\boldsymbol{\theta}} \mathcal{E}[e^2(\boldsymbol{\theta}|y)] \tag{4.48}$$

where $e^2(\boldsymbol{\theta}|y)$ 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 (4.48) 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 pdfs are available but not utilised. In the following example it is shown that the least squared error solution of Equation (4.48) is equivalent to ML solution equation (4.41) 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 $\boldsymbol{y}$ given as

$$\boldsymbol{y} = \boldsymbol{G\theta} + \boldsymbol{e} \tag{4.49}$$

where $\boldsymbol{G}$ is a matrix. The LSE estimate is obtained as the parameter vector at which the gradient of the mean squared error, $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}$, with respect to $\boldsymbol{\theta}$ is zero:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}}\left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G\theta}\right)\Big|_{\theta_{\mathrm{LSE}}} = 0 \tag{4.50}$$

From Equation (4.50) the LSE parameter estimate is given by

$$\boldsymbol{\theta}_{\mathrm{LSE}} = |\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}|^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} \tag{4.51}$$

Note that the MMSE estimate is given by Equation (4.46). Also note that for a Gaussian likelihood function, the LSE solution is the same as the ML solution of Equation (4.41).

4.2.4 Minimum Mean Absolute Value of Error Estimation

The minimum mean absolute value of error (MAVE) estimate (Figure 4.10) is obtained through minimisation of a Bayesian risk function defined as

$$\mathcal{R}_{\mathrm{MAVE}}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right) = \mathcal{E}[(|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|)|\boldsymbol{y}] = \int_{\boldsymbol{\theta}} |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}| f_{\boldsymbol{\theta}|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.52}$$

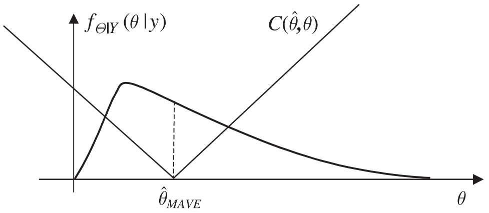


Figure 4.10 Illustration of mean absolute value of error cost function. Note that the MAVE estimate coincides with the conditional median of the posterior function.

In the following it is shown that the minimum mean absolute value estimate is the median of the parameter process. Equation (4.52) can be expressed as

$$\mathcal{R}_{\mathrm{MAVE}}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right) = \int_{-\infty}^{\hat{\theta}} [\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} + \int_{\hat{\theta}}^{\infty} [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}] f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.53}$$

Taking the derivative of the risk function with respect to yields

$$\frac{\partial \mathcal{R}_{\mathrm{MAVE}}\left(\hat{\boldsymbol{\theta}}|\boldsymbol{y}\right)}{\partial \hat{\boldsymbol{\theta}}} = \int_{-\infty}^{\hat{\theta}} f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} - \int_{\hat{\theta}}^{\infty} f_{\Theta|Y}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.54}$$

The minimum absolute value of error is obtained by setting Equation (4.54) to zero:

$$\int_{-\infty}^{\hat{\boldsymbol{\theta}}_{\text{MAVE}}} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\mathbf{y})\, d\boldsymbol{\theta} - \int_{\hat{\boldsymbol{\theta}}_{\text{MAVE}}}^{\infty} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\mathbf{y})\, d\boldsymbol{\theta} \tag{4.55}$$

From Equation (4.55) we note the MAVE estimate is the median of the posterior density.

4.2.5 Equivalence of the MAP, ML, MMSE and MAVE Estimates 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 (4.43), 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.

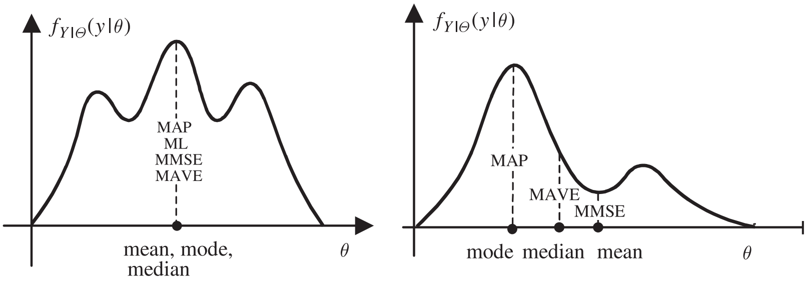


Figure 4.11 Illustration of a symmetric and an asymmetric pdf and their respective mode, mean and median and the relations to MAP, MAVE and MMSE estimates.

4.2.6 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 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 value of a process are compared.

Example 4.6 Estimation of bias and variance of an ML estimator

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

$$y(m) = x(m) + n(m) \quad m = 0, \ldots, N-1 \tag{4.56}$$

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 $\boldsymbol{y}(m)$. The likelihood of an observation vector $\boldsymbol{y} = [y(0), y(1), \ldots, y(N-1)]$ and a parameter value of θ is given by

$$f_{Y|\Theta}\big(\boldsymbol{y}|\theta\big) = \prod_{m=0}^{N-1} f_N\big(y(m)-\theta\big) = \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2\right\} \tag{4.57}$$

From Equation (4.57) the log-likelihood function is given by

$$\ln f_{Y|\Theta}(\boldsymbol{y}|\theta) = -\frac{N}{2}\ln(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2 \tag{4.58}$$

The ML estimate of θ, obtained by setting the derivative of to zero, is given by

$$\hat{\theta}_{ML} = \frac{1}{N}\sum_{m=0}^{N-1} y(m) = \bar{y} \tag{4.59}$$

where $\bar{y}$ denotes the time average of $y(m)$. From Equation (4.59), we note that the ML solution is an unbiased estimate

$$\mathcal{E}[\hat{\theta}_{ML}] = \mathcal{E}\left(\frac{1}{N}\sum_{m=0}^{N-1} y(m)\right) = \mathcal{E}\left(\frac{1}{N}\sum_{m=0}^{N-1}[\theta + n(m)]\right) = \theta \tag{4.60}$$

and the variance of the ML estimate is given by

$$\mathrm{Var}\,[\hat{\theta}_{ML}] = \mathcal{E}[(\hat{\theta}_{ML}-\theta)^2] = \mathcal{E}\left[\left(\frac{1}{N}\sum_{m=0}^{N-1} y(m) - \theta\right)^2\right] = \frac{\sigma_n^2}{N} \tag{4.61}$$

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 a uniform (rectangular-shaped) pdf as

$$f_\Theta(\theta) = \begin{cases} 1/(\theta_{\max} - \theta_{\min}) & \theta_{\min} \le \theta \le \theta_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{4.62}$$

as illustrated in Figure 4.12. From Bayes' rule, the posterior pdf of a uniformly-distributed parameter observed in AWGN is given by

$$\begin{aligned} f_{\Theta|Y}(\theta|\boldsymbol{y}) &= \frac{1}{f_Y(\boldsymbol{y})} f_{Y|\Theta}(\boldsymbol{y}|\theta) f_\Theta(\theta) \\ &= \begin{cases} \dfrac{1}{f_Y(\boldsymbol{y})}\dfrac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left\{-\dfrac{1}{2\sigma_n^2}\sum\limits_{m=0}^{N-1}[y(m)-\theta]^2\right\}, & \theta_{\min} \le \theta \le \theta_{\max} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4.63}$$

The MAP estimate is obtained by maximising the posterior pdf:

$$\hat{\theta}_{\mathrm{MAP}}(\boldsymbol{y}) = \begin{cases} \theta_{\min} & \text{if} \quad \hat{\theta}_{ML}(\boldsymbol{y}) < \theta_{\min} \\ \hat{\theta}_{ML}(\boldsymbol{y}) & \text{if} \quad \theta_{\min} \le \hat{\theta}_{ML}(\boldsymbol{y}) \le \theta_{\max} \\ \theta_{\max} & \text{if} \quad \hat{\theta}_{ML}(\boldsymbol{y}) > \theta_{\max} \end{cases} \tag{4.64}$$

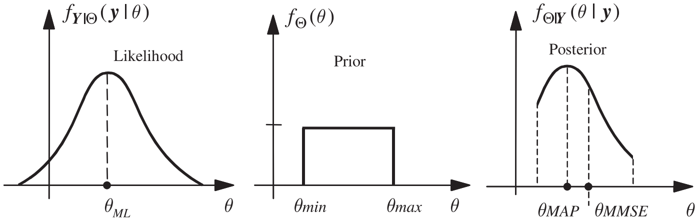


Figure 4.12 Illustration of the effects of a uniform prior on the estimate of a parameter observed in AWGN, where it is assumed that $\theta_{\min} \le \theta_{ML} \le \theta_{\max}$.

Note that the MAP estimate is constrained to the range $\theta_{\min}$ to $\theta_{\max}$. This constraint is desirable and moderates the estimates that, due to say low signal-to-noise ratio, fall outside the range of possible values of θ. It is easy to see that the variance of an estimate constrained to a range of $\theta_{\min}$ to $\theta_{\max}$ is less than the variance of the ML estimate in which there is no constraint on the range of the parameter estimate:

$$\mathrm{Var}[\hat{\theta}_{\mathrm{MAP}}] = \int_{\boldsymbol{\theta}_{min}}^{\theta_{\max}} (\hat{\theta}_{\mathrm{MAP}} - \theta)^2 f_{Y|\Theta}(\mathbf{y}|\theta)d\theta \le \mathrm{Var}[\hat{\theta}_{\mathrm{ML}}] = \int_{-\infty}^{\infty} (\hat{\theta}_{\mathrm{ML}} - \theta)^2 f_{Y|\Theta}(\mathbf{y}|\theta)d\theta \tag{4.65}$$

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 as

$$f_\Theta(\theta) = \frac{1}{(2\pi\sigma_\theta^2)^{1/2}} \exp\left(-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}\right) \tag{4.66}$$

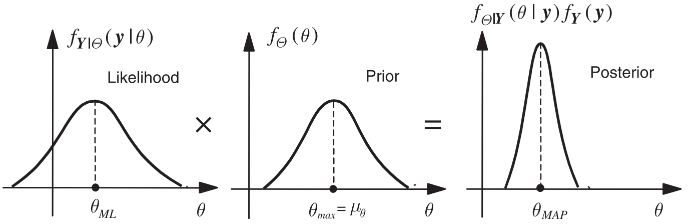

$f_{Y|\Theta}(\mathbf{y}\,|\,\theta)$
Likelihood
θ_{ML}
θ
×
$f_\Theta(\theta)$
Prior
$\theta_{max} = \mu_\theta$
θ
=
$f_{\Theta|Y}(\theta\,|\,\mathbf{y})f_Y(\mathbf{y})$
Posterior
θ_{MAP}
θ

Figure 4.13 Illustration of the posterior pdf as product of the likelihood and the prior.

From Bayes' rule the posterior pdf is given as the product of the likelihood and the prior pdfs as

$$\begin{aligned} f_{\Theta|Y}(\theta|\mathbf{y}) &= \frac{1}{f_Y(\mathbf{y})} f_{Y|\Theta}(\mathbf{y}|\theta) f_\Theta(\theta) \\ &= \frac{1}{f_Y(\mathbf{y})} \frac{1}{(2\pi\sigma_n^2)^{N/2}(2\pi\sigma_\theta^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2 - \frac{1}{2\sigma_\theta^2}(\theta-\mu_\theta)^2\right\} \end{aligned} \tag{4.67}$$

The maximum posterior solution is obtained by setting the derivative of the log-posterior function, with respect to θ to zero:

$$\hat{\theta}_{MAP}(\mathbf{y}) = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N}\bar{y} + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N}\mu_\theta \tag{4.68}$$

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 4.13. Also note that in Equation (4.68) 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 observations 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 (4.68) is the term $\bar{y}$, and that E $[\bar{y}] = \theta$

$$\mathcal{E}[\hat{\theta}_{\text{MAP}}(\boldsymbol{y})] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N}\theta + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N}\mu_\theta \tag{4.69}$$

and the variance of the MAP estimate is given as

$$\text{Var}\left[\hat{\theta}_{\text{MAP}}(\boldsymbol{y})\right] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \times \text{Var}[\bar{y}] = \frac{\sigma_n^2/N}{1 + \sigma_n^2/N\sigma_\theta^2} \tag{4.70}$$

Substitution of Equation (4.61) in Equation (4.70) yields

$$\text{Var}[\hat{\theta}_{\text{MAP}}(\boldsymbol{y})] = \frac{\text{Var}[\hat{\theta}_{\text{ML}}(\boldsymbol{y})]}{1 + \text{Var}[\hat{\theta}_{\text{ML}}(\boldsymbol{y})]/\sigma_\theta^2} \tag{4.71}$$

Note that as 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 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 (4.68) 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:

$$\underset{N\to\infty}{\text{limit}}\,\hat{\theta}_{MAP}(\boldsymbol{y}) = \underset{N\to\infty}{\text{limit}}\left(\frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N}\bar{y} + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N}\mu_\theta\right) = \bar{y} = \hat{\theta}_{ML} \tag{4.72}$$

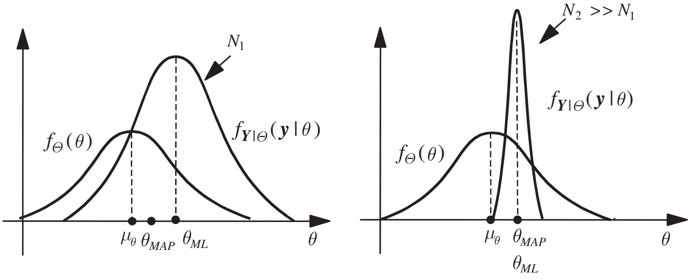

N_1
$f_\Theta(\theta)$
$f_{Y|\Theta}(\boldsymbol{y}|\theta)$
μ_θ θ_{MAP} θ_{ML}
θ
$N_2 >> N_1$
$f_{Y|\Theta}(\boldsymbol{y}|\theta)$
$f_\Theta(\theta)$
μ_θ θ_{MAP}
θ_{ML}
θ

Figure 4.14 Illustration of the effect of the increasing length of observation on the variance an estimator.

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 scalar Gaussian 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

$$y(m) = x(m) + n(m) \tag{4.73}$$

The posterior pdf of the signal $x(m)$ is given by

$$\begin{aligned} f_{X|Y}\big(x(m)|y(m)\big) &= \frac{1}{f_Y\big(y(m)\big)} f_{Y|X}\big(y(m)|x(m)\big) f_X\big(x(m)\big) \\ &= \frac{1}{f_Y\big(y(m)\big)} \underbrace{f_N\big(y(m)-x(m)\big)}_{\text{Likelihood}} \underbrace{f_X\big(x(m)\big)}_{\text{Prior}} \end{aligned} \tag{4.74}$$

where $f_X(x(m)) = \mathcal{N}\big(x(m), \mu_x, \sigma_x^2\big)$ and $f_N(n(m)) = \mathcal{N}\big(n(m), \mu_n, \sigma_n^2\big)$ are the Gaussian pdfs of the signal and noise respectively. Substitution of the signal and noise pdfs in Equation (4.74) yields

$$\begin{aligned} f_{X|Y}\big(x(m)|y(m)\big) = \frac{1}{f_Y\big(y(m)\big)} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{-\frac{[y(m)-x(m)-\mu_n]^2}{2\sigma_n^2}\right\} \\ \times \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left\{-\frac{[x(m)-\mu_x]^2}{2\sigma_x^2}\right\} \end{aligned} \tag{4.75}$$

This equation can be rewritten as

$$\begin{aligned} f_{X|Y}\big(x(m)|y(m)\big) = \frac{1}{f_Y\big(y(m)\big)} \frac{1}{2\pi\sigma_n\sigma_x} \times \\ \exp\left\{-\frac{\sigma_x^2[y(m)-x(m)-\mu_n]^2 + \sigma_n^2[x(m)-\mu_x]^2}{2\sigma_x^2\sigma_n^2}\right\} \end{aligned} \tag{4.76}$$

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 the estimate $\hat{x}(m)$ to zero as

$$\frac{\partial\left[\ln f_{X|Y}\big(x(m)|y(m)\big)\right]}{\partial\hat{x}(m)} = -\frac{-2\sigma_x^2[y(m)-x(m)-\mu_n] + 2\sigma_n^2[x(m)-\mu_x]}{2\sigma_x^2\sigma_n^2} = 0 \tag{4.77}$$

From Equation (4.77) the MAP signal estimate is given by

$$\hat{x}(m) = \frac{\sigma_x^2}{\sigma_x^2+\sigma_n^2}[y(m)-\mu_n] + \frac{\sigma_n^2}{\sigma_x^2+\sigma_n^2}\mu_x \tag{4.78}$$

Note that the estimate $\hat{x}(m)$ is a weighted linear interpolation between the unconditional mean of $x(m)$, $\boldsymbol{\mu_x}$, and the observed value $(y(m) - \boldsymbol{\mu_n})$. At a very poor SNR, i.e. when $\sigma_x^2 << \sigma_n^2$ we have $\hat{x}(m) \approx \mu_x$; and, on the other hand, for a noise-free signal $\sigma_n^2 = 0$ and $\boldsymbol{\mu_n} = 0$ and we have $\hat{x}(m) = y(m)$.

Example 4.10 MAP estimate of a Gaussian – AR process observed in AWGN

Consider a vector $\boldsymbol{x}$ of N samples from an autoregressive (AR) process observed in an additive Gaussian noise, and modelled as

$$y = x + n \tag{4.79}$$

From Chapter 8, a vector $\boldsymbol{x}$ from an AR process may be expressed as

$$\boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} \tag{4.80}$$

where $\boldsymbol{A}$ is a matrix of the AR model coefficients, and the vector $\boldsymbol{e}$ is the input signal of the AR model. Assuming that the signal $\boldsymbol{x}$ is Gaussian, and that the P initial samples $\boldsymbol{x}_0$ are known, the pdf of the signal $\boldsymbol{x}$ is given by

$$f_X(\boldsymbol{x}|\boldsymbol{x}_0) = f_E(\boldsymbol{e}) = \frac{1}{\left(2\pi\,\sigma_e^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \tag{4.81}$$

where it is assumed that the input signal $\boldsymbol{e}$ of the AR model is a zero-mean uncorrelated process with variance σ_e^2. The pdf of a zero-mean Gaussian noise vector $\boldsymbol{n}$, with covariance matrix $\boldsymbol{\Sigma}_{nn}$, is given by

$$f_N(\boldsymbol{n}) = \frac{1}{(2\pi)^{N/2}\,|\boldsymbol{\Sigma}_{nn}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{\Sigma}_{nn}^{-1}\boldsymbol{n}\right) \tag{4.82}$$

From Bayes' rule, the pdf of the signal given the noisy observation is

$$f_{X|Y}(\boldsymbol{x}|\boldsymbol{y}) = \frac{f_{Y|X}(\boldsymbol{y}\,|\boldsymbol{x})f_X(\boldsymbol{x})}{f_Y(\boldsymbol{y})} = \frac{1}{f_Y(\boldsymbol{y})} f_N(\boldsymbol{y}-\boldsymbol{x}) f_X(\boldsymbol{x}) \tag{4.83}$$

Substitution of the pdfs of the signal and noise in Equation (4.83) yields

$$f_{X|Y}(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{f_Y(\boldsymbol{y})\,(2\pi)^N \sigma_e^{N/2}\,|\boldsymbol{\Sigma}_{nn}|^{1/2}} \exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{nn}^{-1}(\boldsymbol{y}-\boldsymbol{x}) + \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\sigma_e^2}\right]\right\} \tag{4.84}$$

The MAP estimate corresponds to the minimum of the argument of the exponential function in Equation (4.84). Assuming that the argument of the exponential function is differentiable, and has a well-defined minimum, we can obtain the MAP estimate from

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) = \arg\underset{\boldsymbol{x}}{\text{zero}}\left\{\frac{\partial}{\partial\,\boldsymbol{x}}\left[(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{nn}^{-1}(\boldsymbol{y}-\boldsymbol{x}) + \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\sigma_e^2}\right]\right\} \tag{4.85}$$

The MAP estimate is

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}}(\boldsymbol{y}) = \left(\mathbf{I} + \frac{1}{\sigma_e^2}\boldsymbol{\Sigma}_{nn}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{y} \tag{4.86}$$

where $\mathbf{I}$ is the identity matrix.

4.3 Expectation-Maximisation (EM) Method

The EM algorithm (EM stands for expectation-maximisation or alternatively estimate-maximise) is used for calculation of the maximum likelihood (ML) estimate of a parameter vector $\boldsymbol{\theta}$ given an incomplete related observation $\boldsymbol{y}$. The EM algorithm is an iterative ascent, 'hill climbing', method for finding the solution that maximises the expectation of a likelihood function $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$. It has applications in blind deconvolution, clustering, training of hidden Markov models, model-based signal interpolation, spectral estimation from noisy observations and signal restoration.

4.3.1 Complete and Incomplete Data

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 data samples as in signal restoration or missing data labels as in unsupervised clustering problems, or because the

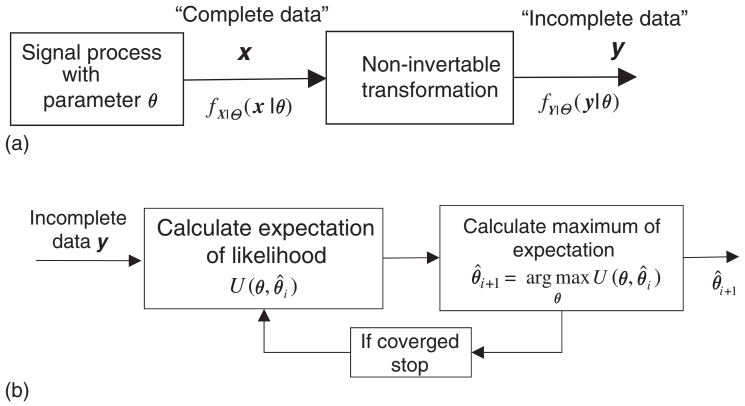

"Complete data"
"Incomplete data"
Signal process with parameter θ
$\boldsymbol{x}$
$f_{X|\Theta}(\boldsymbol{x}|\theta)$
Non-invertable transformation
$\boldsymbol{y}$
$f_{Y|\Theta}(\boldsymbol{y}|\theta)$
(a)
Incomplete data $\boldsymbol{y}$
Calculate expectation of likelihood
$U(\theta, \hat{\theta}_i)$
Calculate maximum of expectation
$\hat{\theta}_{i+1} = \arg\max_{\theta} U(\theta, \hat{\theta}_i)$
$\hat{\theta}_{i+1}$
If coverged stop
(b)

Figure 4.15 Illustration of (a) transformation of complete data, signal process $\boldsymbol{x}$ with hidden parameter $\boldsymbol{\theta}$, to incomplete data $\boldsymbol{y}$, (b) EM estimation of hidden parameter $\boldsymbol{\theta}$ from incomplete data $\boldsymbol{y}$.

problem is difficult. Hence, instead of a direct ML estimate, the expectation of the ML estimate is maximised.

For example, in clustering applications the complete data are the observations with the labels attached. Supervised clustering methods use the complete data. However, usually the raw observation data are incomplete as they do not have a cluster label attached to each and hence an iterative EM process is employed consisting of the following two stages: (a) labelling of the data with cluster as in $\boldsymbol{x}_j$; $\boldsymbol{x}$ belonging to cluster j and (b) calculation of means and variances of clusters. The iterations are continued until a convergence criterion is satisfied. Most signal processing and machine-learning clustering tasks involve unsupervised clustering where initially the signals do not have a cluster label.

To formally define the notions of complete and incomplete data, consider a signal $\boldsymbol{x}$ from a random process $\boldsymbol{X}$ with an unknown latent (hidden) parameter vector $\boldsymbol{\theta}$ and a joint pdf $f_{X,\Theta}(\boldsymbol{x},\boldsymbol{\theta})$, Figure 4.15. The signal $\boldsymbol{x}$ is the so-called complete data and the ML estimate of the parameter vector $\boldsymbol{\theta}$ may be obtained from the likelihood function $f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})$. Now assume that the signal $\boldsymbol{x}$ goes through a many-to-one non-invertible transformation (e.g. when a number of samples of the vector $\boldsymbol{x}$ are lost or when the signal labels are unavailable) and is observed as $\boldsymbol{y}$. The observation $\boldsymbol{y}$ is then the so-called incomplete data.

4.3.2 Maximisation of Expectation of the Likelihood Function

Maximisation of the likelihood of the incomplete data, $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ with respect to the parameter vector $\boldsymbol{\theta}$ is often a difficult task, whereas maximisation of the likelihood of the complete data $f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})$ 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 given the observation, $\left[\ln f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})|\boldsymbol{y}\right]$, expressed as

$$\mathcal{E}\left[\ln f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})|\boldsymbol{y}\right] = \int_{X} f_{X|Y,\Theta}\,(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \ln f_{X|\Theta}\,(\boldsymbol{x}|\boldsymbol{\theta})\; d\boldsymbol{x} \tag{4.87}$$

In Equation (4.87), the computation of the term $f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$ requires an estimate of the unknown parameter vector $\boldsymbol{\theta}$. For this reason, the expectation of the likelihood function is maximised iteratively starting with an initial estimate of $\boldsymbol{\theta}$ and updating the estimate as described in the following. Note that the right hand side of Equation (4.87) is similar to an entropy function.

EM Algorithm

Step 1: Initialisation – Select an initial parameter estimate $\boldsymbol{\theta}_0$, and
For $i = 0,\ 1, \ldots$ until convergence:

Step 2: Expectation – Compute expectation of log likelihood of $\boldsymbol{\theta}$

$$\begin{aligned} U(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) &= E[\ln f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i] \\ &= \int_X f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i) \ln f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x} \end{aligned} \tag{4.88}$$

Step 3: Maximisation – Select

$$\hat{\boldsymbol{\theta}}_{i+1} = \arg\max_{\boldsymbol{\theta}} U(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) \tag{4.89}$$

Step 4: Convergence test – If not converged then go to Step 2.

4.3.3 Derivation and 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 $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$. The joint pdf of the complete data $\boldsymbol{x}$ and the incomplete observation $\boldsymbol{y}$ conditioned on parameter value $\boldsymbol{\theta}$ can be written as

$$f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}) f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) \tag{4.90}$$

where $f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$ is the pdf of $\boldsymbol{x}$ and $\boldsymbol{y}$ with $\boldsymbol{\theta}$ as a parameter. From Equation (4.90), the log-likelihood of the incomplete data is obtained as

$$\ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) = \ln f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) - \ln f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}) \tag{4.91}$$

Using an estimate of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_i$, and taking the expectation of Equation (4.91) – this is done by multiplying both sides of Equation (4.91) by $f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i)$ and integrating w.r.t. $\boldsymbol{x}$ – we have

$$\ln f_{Y,\Theta}(\boldsymbol{y}|\boldsymbol{\theta}) = U(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) - V(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) \tag{4.92}$$

Since expectation is taken w.r.t. $\boldsymbol{x}$, $\ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ is unaffected, i.e. $\mathcal{E}[\ln f_{Y|\Theta}(\mathbf{y}|\boldsymbol{\theta})]_{\text{w.r.t } \mathbf{x}} = \ln f_{Y|\Theta}(\mathbf{y}|\boldsymbol{\theta})]$.

The function $U(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ is the conditional expectation of $\ln f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})$, conditioned on $\boldsymbol{y}$ and the current estimate $\hat{\boldsymbol{\theta}}_i$ and defined as

$$\begin{aligned} U(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_i) &= \mathcal{E}[\ln f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i] \\ &= \int_X f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i) \ln f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta}) d\boldsymbol{x} \end{aligned} \tag{4.93}$$

Note that $f_{X,Y|\Theta}(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = f_{X|\Theta}(\boldsymbol{x}|\boldsymbol{\theta})$ as the complete information $\boldsymbol{x}$ includes the incomplete information $\boldsymbol{y}$.

The function $V(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i)$ is the conditional expectation of $\ln f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta})$, conditioned on $\boldsymbol{y}$ and the current estimate $\hat{\boldsymbol{\theta}}_i$ and defined as

$$\begin{aligned} V(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_i) &= \mathcal{E}\left[\ln f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta})|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i\right] \\ &= \int_X f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \hat{\boldsymbol{\theta}}_i) \ln f_{X|Y,\Theta}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta})\ d\boldsymbol{x} \end{aligned} \tag{4.94}$$

Now, from Equation (4.92), the log-likelihood of the incomplete data $\boldsymbol{y}$ with parameter estimate at iteration i is given by

$$\ln f_{Y,\boldsymbol{\Theta}}(\boldsymbol{y},\hat{\boldsymbol{\theta}}_i) = U\left(\hat{\boldsymbol{\theta}}_i,\hat{\boldsymbol{\theta}}_i\right) - V\left(\hat{\boldsymbol{\theta}}_i,\hat{\boldsymbol{\theta}}_i\right) \tag{4.95}$$

It can be shown (see Dempster et al., 1977) that the function V satisfies the inequality

$$V(\hat{\boldsymbol{\theta}}_{i+1},\hat{\boldsymbol{\theta}}_i) \le V(\hat{\boldsymbol{\theta}}_i,\hat{\boldsymbol{\theta}}_i) \tag{4.96}$$

and in the maximisation of the expectation of $\ln f_{X|\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{\theta})$ we choose $\hat{\boldsymbol{\theta}}_{i+1}$ such that

$$U(\hat{\boldsymbol{\theta}}_{i+1},\hat{\boldsymbol{\theta}}_i) \ge U(\hat{\boldsymbol{\theta}}_i,\hat{\boldsymbol{\theta}}_i) \tag{4.97}$$

From Equation (4.95) and the inequalities (4.96) and (4.97), it follows that

$$\ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_{i+1}) \ge \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\hat{\boldsymbol{\theta}}_i) \tag{4.98}$$

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 $\ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$.

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 $\hat{\theta}_i$, with different values of the observation signal $\boldsymbol{y}$ and the actual parameter vector $\boldsymbol{\theta}$. The minimum estimation variance depends on the distributions of the parameter vector $\boldsymbol{\theta}$ and on the observation signal $\boldsymbol{y}$. 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 the estimate of the i^{th} coefficient θ_i of a parameter vector $\boldsymbol{\theta}$ is given as

$$\mathrm{Var}\left[\hat{\theta}_i(\boldsymbol{y})\right] \ge \frac{\left(1+\dfrac{\partial\theta_{Bias}}{\partial\theta_i}\right)^2}{\underbrace{\mathcal{E}\left[\left(\dfrac{\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\theta_i}\right)^2\right]}_{J_{ii}(\boldsymbol{\theta})}} \tag{4.99}$$

An estimator that achieves the lower bound on the variance is called the minimum variance, or the most efficient, estimator.

Interpretation – In Equation (4.99), the expectation of the second derivative of the log-likelihood function $\mathcal{E}\left[\left(\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})/\partial\theta_i\right)^2\right] = [\boldsymbol{J}(\boldsymbol{\theta})]_{ii}$ is the Fisher information metric whose magnitude increases with the increasing slope or peakiness (decreasing variance) of the pdf. Equation (4.99) indicates that, for an unbiased estimator or when $\partial\theta_{Bias}/\partial\theta_i = 0$, the variance of the estimate $\mathrm{Var}[\hat{\theta}_i(\boldsymbol{y})]$ is bounded by the reciprocal of $[\boldsymbol{J}(\boldsymbol{\theta})]_{ii}$. Hence, as expected, a likelihood function with a steeper slope, indicating a smaller variance and a more peaky shape, results in a smaller bound on the variance of the estimate.

Proof – The bias, or mean of the error, in the estimate of the i^{th} coefficient of the parameter vector $\boldsymbol{\theta}$, averaged over the observation space $\boldsymbol{Y}$ is defined as

$$E[\hat{\theta}_i(\boldsymbol{y}) - \theta_i] = \int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_i] f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} = \theta_{Bias} \tag{4.100}$$

Differentiation of Equation (4.100) with respect to θ_i yields

$$\int_{-\infty}^{\infty} \left\{ [\hat{\theta}_i(\boldsymbol{y}) - \theta_i] \frac{\partial\, f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\, \theta_i} - f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \right\} d\boldsymbol{y} = \frac{\partial\, \theta_{Bias}}{\partial\, \theta_i} \tag{4.101}$$

For a probability density function we have

$$\int_{-\infty}^{\infty} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} = 1 \tag{4.102}$$

Therefore Equation (4.101) can be written as

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_i] \frac{\partial\, f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\, \theta_i}\, d\boldsymbol{y} = 1 + \frac{\partial\, \theta_{Bias}}{\partial\, \theta_i} \tag{4.103}$$

Now, since the derivative of the integral of pdf equation (4.102) is zero, taking the derivative of Equation (4.102) and multiplying the result by θ_{Bias} yields

$$\theta_{Bias} \int_{-\infty}^{\infty} \frac{\partial f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\, \theta_i}\, d\boldsymbol{y} = 0 \tag{4.104}$$

Substituting $\partial f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})/\partial\theta_i = f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})/\partial\theta_i$ into Equation (4.103), and using Equation (4.104), we obtain

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_{Bias} - \theta_i] \frac{\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\, \theta_i} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} = 1 + \frac{\partial \theta_{Bias}}{\partial \theta_i} \tag{4.105}$$

Now squaring both sides of Equation (4.105), we obtain

$$\left(\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_{Bias} - \theta_i] \frac{\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\, \theta_i} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} \right)^2 = \left(1 + \frac{\partial \theta_{Bias}}{\partial \theta_i} \right)^2 \tag{4.106}$$

For the left-hand side of Equation (4.106) application of the Schwartz inequality

$$\left(\int_{-\infty}^{\infty} f(y) g(y)\, dy \right)^2 \le \int_{-\infty}^{\infty} (f(y))^2\, dy \times \int_{-\infty}^{\infty} (g(y))^2\, dy \tag{4.107}$$

yields

$$\begin{aligned} &\left\{ \int_{-\infty}^{\infty} \left([\hat{\theta}_i(\boldsymbol{y}) - \theta_{Bias} - \theta_i] f^{1/2}_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \right) \left(\frac{\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} f^{1/2}_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \right) d\boldsymbol{y} \right\}^2 \le \\ &\left\{ \int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_{Bias} - \theta_i]^2 f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} \right\} \left\{ \int_{-\infty}^{\infty} \left(\frac{\partial \ln f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} \right)^2 f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} \right\} \end{aligned} \tag{4.108}$$

From Equations (4.106) and (4.108), we have

$$\mathrm{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq \frac{\left(1+\dfrac{\partial \theta_{Bias}}{\partial\ \theta_i}\right)^2}{\mathcal{E}\left[\left(\dfrac{\partial \ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial\ \theta_i}\right)^2\right]} \tag{4.109}$$

The Cramer–Rao inequality (4.99) results directly from the inequality (4.109).

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 (4.99) instead of the likelihood $f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})$ we use the joint pdf $f_{Y,\Theta}(\boldsymbol{y},\ \boldsymbol{\theta})$, and we also use the logarithmic relation

$$\frac{\partial \ln f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial \ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \ln f_{\Theta}(\boldsymbol{\theta})}{\partial \theta_i} \tag{4.110}$$

The Cramer–Rao bound for random parameters is obtained as

$$\mathrm{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq \frac{\left(1+\dfrac{\partial\ \theta_{Bias}}{\partial\ \theta_i}\right)^2}{\mathcal{E}\left[\left(\dfrac{\partial\ \ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i}\right)^2 + \left(\dfrac{\partial\ \ln f_{\Theta}(\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right]} \tag{4.111}$$

where the second term in the denominator of Equation (4.111) describes the effect of the prior pdf of $\boldsymbol{\theta}$. As expected the use of the prior $f_{\Theta}(\boldsymbol{\theta})$ can result in a decrease in the variance of the estimate. An alternative form of the minimum bound on estimation variance can be obtained by using the likelihood relation

$$\mathcal{E}\left[\left(\frac{\partial\ \ln f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})}{\partial\ \theta_i}\right)^2\right] = -\mathcal{E}\left[\frac{\partial^2 \ln f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})}{\partial\ \theta_i^2}\right] \tag{4.112}$$

as

$$\mathrm{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq -\frac{\left(1+\dfrac{\partial\ \theta_{Bias}}{\partial\ \theta_i}\right)^2}{\mathcal{E}\left[\dfrac{\partial^2\ \ln f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i^2} + \dfrac{\partial^2\ \ln f_{\Theta}(\boldsymbol{\theta})}{\partial \theta_i^2}\right]} \tag{4.113}$$

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

$$\mathrm{Cov}\,[\hat{\boldsymbol{\theta}}] \geq \boldsymbol{J}^{-1}(\boldsymbol{\theta}) \tag{4.114}$$

where $\boldsymbol{J}$ is the Fisher information matrix, with its elements given by

$$[\boldsymbol{J}(\boldsymbol{\theta})]_{ij} = -\mathcal{E}\left[\frac{\partial^2\ \ln f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\right] \tag{4.115}$$

The lower bound on the variance of the i^{th} element of the vector $\boldsymbol{\theta}$ is given by

$$\mathrm{Var}(\hat{\theta}_i) \geq \left[\boldsymbol{J}^{-1}(\boldsymbol{\theta})\right]_{ii} = -\frac{1}{\mathcal{E}\left[\dfrac{\partial^2 \ln f_{Y,\Theta}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i^2}\right]} \tag{4.116}$$

where $\left[\boldsymbol{J}^{-1}(\boldsymbol{\theta})\right]_{ii}$ is the i^{th} diagonal element of the inverse of the Fisher matrix.

4.5 Design of Gaussian Mixture Models (GMMs)

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 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.

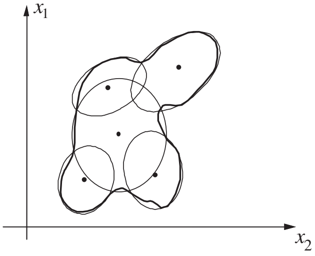


Figure 4.16 Illustration of probabilistic modelling of a two-dimensional signal space with a mixture of five bivariate Gaussian densities.

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 $\boldsymbol{X}$ is defined as

$$f_X(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.117}$$

where $\mathcal{N}_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the k^{th} component of the Gaussian mixture 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 $\boldsymbol{X}_k$ 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 expectation–maximisation (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 4.3, is an iterative maximum-likelihood (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 $[\boldsymbol{y}(m),\ m=0,\ldots,N-1]$ form the incomplete data. The complete data may be viewed as the observation vectors with a label attached to each vector $\boldsymbol{y}(m)$ to indicate the component of the mixture Gaussian model that generated the vector. Note that if each signal vector $\boldsymbol{y}(m)$ 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 $\boldsymbol{y}(m),\ m=0,\ldots,N-1$
The complete data $\boldsymbol{x}(m)=[\boldsymbol{y}(m),k]=\boldsymbol{y}_k(m),\ m=0,\ldots,N-1, k\in(1,\ldots,\ K)$

The probability of the complete data is the probability that an observation vector $\boldsymbol{y}(m)$ 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

$$\begin{aligned}U(\boldsymbol{\Theta},\hat{\boldsymbol{\Theta}}_{\iota}) &= \mathcal{E}[\ln f_{Y,K;\boldsymbol{\Theta}}(\boldsymbol{y}(m),k;\boldsymbol{\Theta})|\boldsymbol{y}(m);\hat{\boldsymbol{\Theta}}_i] \\ &= \sum_{m=0}^{N-1}\sum_{k=1}^{K}\frac{f_{Y,K|\boldsymbol{\Theta}}(\boldsymbol{y}(m),k|\,\hat{\boldsymbol{\Theta}}_i)}{f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}\ln f_{Y,K;\boldsymbol{\Theta}}(\boldsymbol{y}(m),k;\boldsymbol{\Theta})\end{aligned} \tag{4.118}$$

where $\boldsymbol{\Theta}=\{\boldsymbol{\theta}_k=[P_k,\ \boldsymbol{\mu}_k,\ \boldsymbol{\Sigma}_k],\ k=1,\ldots,\ K\}$, are the parameters of the Gaussian mixture as in Equation (4.117). Now the joint pdf of $\boldsymbol{y}(m)$ and the k^{th} Gaussian component of the mixture density can be written as

$$\begin{aligned}f_{Y,K\,|\boldsymbol{\Theta}}\left(\boldsymbol{y}(m)\,,k\left|\hat{\boldsymbol{\theta}}_i\right.\right) &= P_{k_i}f_k\left(\boldsymbol{y}(m)\left|\hat{\boldsymbol{\theta}}_{k_i}\right.\right) \\ &= P_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right)\end{aligned} \tag{4.119}$$

where $\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right)$ is a Gaussian density with mean vector and covariance matrix:

$$\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\Sigma}}_k,\hat{\boldsymbol{\mu}}_k\right)=\frac{1}{(2\pi)^{P/2}|\hat{\boldsymbol{\Sigma}}_k|^{1/2}}\exp\left\{-\frac{1}{2}\left(\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_k\right)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_k^{-1}\left(\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_k\right)\right\} \tag{4.120}$$

The pdf of $\boldsymbol{y}(m)$ as a mixture of K Gaussian densities is given by

$$\begin{aligned}f_{Y|\boldsymbol{\theta}}\left(\boldsymbol{y}(m)\left|\hat{\boldsymbol{\theta}}_i\right.\right) &= \mathcal{N}\left(\boldsymbol{y}(m)\left|\hat{\boldsymbol{\theta}}_i\right.\right) \\ &= \sum_{k=1}^{K}\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right)\end{aligned} \tag{4.121}$$

Substitution of the Gaussian densities of Equation (4.119) and Equation (4.121) in Equation (4.118) yields

$$\begin{aligned}U\left[(\boldsymbol{\mu};\boldsymbol{\Sigma},\boldsymbol{P}),(\hat{\boldsymbol{\mu}}_i,\hat{\boldsymbol{\Sigma}}_i,\ \hat{\boldsymbol{P}}_i)\right] &= \sum_{m=0}^{N-1}\sum_{k=1}^{K}\frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}\ln\left[P_k\mathcal{N}_k\left(\boldsymbol{y}(m);\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\right] \\ &= \sum_{m=0}^{N-1}\sum_{k=1}^{K}\left(\frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}\ln P_{k_i}+\frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}\ln\mathcal{N}\left(\boldsymbol{y}(m);\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\right)\end{aligned} \tag{4.122}$$

Equation (4.122) is maximised with respect to the parameter P_k using the constrained optimisation method. This involves subtracting the constant term $\Sigma P_k = 1$ from the right-hand side of Equation (4.122) and then setting the derivative of this equation with respect to P_k to zero; this yields

$$\hat{P}_{k_{i+1}} = \underset{P_k}{\arg\max}\ U\left[(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \boldsymbol{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)\right] = \frac{1}{N}\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)} \tag{4.123}$$

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:

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{k_{i+1}} &= \underset{\hat{\mu}_k}{\arg\max}\ U\left[(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \boldsymbol{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)\right] \\ &= \frac{\displaystyle\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}\boldsymbol{y}(m)}{\displaystyle\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right)}{\mathcal{N}(\boldsymbol{y}(m)|\,\hat{\boldsymbol{\Theta}}_i)}}\end{aligned} \tag{4.124}$$

and

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_{k_{i+1}} &= \underset{\hat{\Sigma}_k}{\arg\max}\ U\left[(\boldsymbol{\mu}; \boldsymbol{\Sigma}, \boldsymbol{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)\right] \\ &= \frac{\displaystyle\sum_{m=0}^{N-1} \frac{\hat{P}_{k_1}\mathcal{N}_k(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_1}, \hat{\boldsymbol{\Sigma}}_{k_1})}{\mathcal{N}(\boldsymbol{y}(m)|\hat{\boldsymbol{\Theta}}_i)}\left(\boldsymbol{y}(m) - \hat{\boldsymbol{\mu}}_{k_i}\right)\left(\boldsymbol{y}(m) - \hat{\boldsymbol{\mu}}_{k_i}\right)^{\mathrm{T}}}{\displaystyle\sum_{m=0}^{N-1} \frac{\hat{P}_{k_1}\mathcal{N}_k(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}(\boldsymbol{y}(m)|\hat{\boldsymbol{\Theta}}_i)}}\end{aligned} \tag{4.125}$$

Equations (4.123)–(4.125) 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 $\{\boldsymbol{y}(m)\}$ with one of M classes of signals $\{C_k;\ k = 1, \ldots, M\}$ 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 systems that are based on acoustic models of phonemes, each speech segment is labelled with one of the phonemes; and in speech or video compression, a segment of speech samples or a block of image pixels are 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 $\boldsymbol{x}$ could belong to either of the two classes. The shaded area gives a measure of the classification error. The obvious

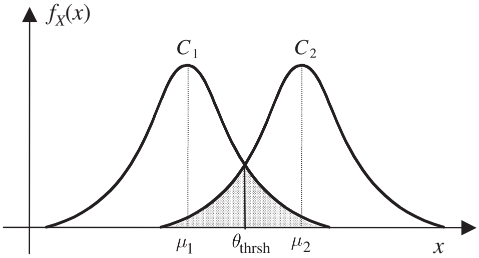


Figure 4.17 Illustration of the overlap of the distribution of two classes of signals.

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 telecommunication 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 characterise each class adequately 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 4.18(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 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 $\boldsymbol{x}$ is labelled with the class that scores the higher a posterior probability:

$$P_{C|X}\left(C_1\,|\boldsymbol{x}\right) \underset{C_2}{\overset{C_1}{\lessgtr}} P_{C|X}\left(C_2\,|\boldsymbol{x}\right) \tag{4.126}$$

Note the above notation means that a signal $\boldsymbol{x}$ is classified as C_1 if $P_{C|X}(C_1|\boldsymbol{x}) > P_{C|X}(C_2|\boldsymbol{x})$ otherwise it is classified as C_2. Using Bayes' rule Equation (4.126) can be rewritten as

$$P_C\left(C_1\right) f_{X|C}\left(\boldsymbol{x}\,|C_1\right) \underset{C_2}{\overset{C_1}{\lessgtr}} P_C\left(C_2\right) f_{X|C}\left(x\,|C_2\right) \tag{4.127}$$

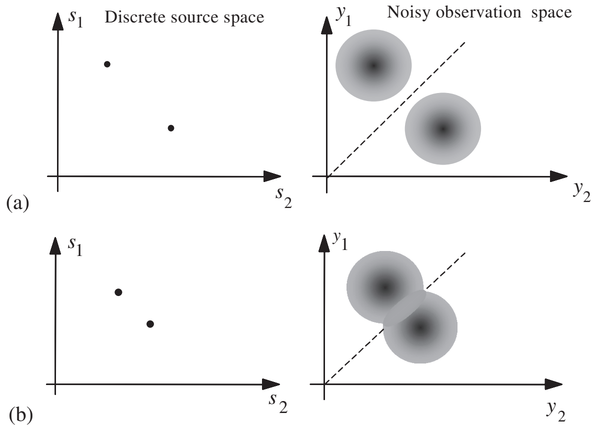


Figure 4.18 Illustration of binary classification: (a) the source and observation spaces are well separated; (b) the observation spaces overlap.

Letting $P_C(C_1) = P_1$ and $P_C(C_2) = P_2$, Equation (4.127) is often written in terms of a likelihood ratio test as

$$\frac{f_{X|C}\ (\boldsymbol{x}\,|C_1\,)}{f_{X|C}\ (\boldsymbol{x}\,|C_2\,)} \underset{C_2}{\overset{C_1}{\gtrless}} \frac{P_2}{P_1} \tag{4.128}$$

Taking the likelihood ratio yields the discriminant function

$$h(\boldsymbol{x}) = \ln f_{X|C}\ (\boldsymbol{x}\,|C_1\,) - \ln f_{X|C}\ (\boldsymbol{x}\,|C_2\,) \underset{C_2}{\overset{C_1}{\gtrless}} \ln \frac{P_2}{P_1} \tag{4.129}$$

Now assume that the signal in each class has a Gaussian distribution with a probability distribution function given by

$$f_{X|C}\ (\boldsymbol{x}\,|c_i\,) = \frac{1}{\sqrt{2\pi}\ |\boldsymbol{\Sigma}_i|} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right], \quad i = 1,2 \tag{4.130}$$

From Equations (4.129) and (4.130), the discriminant function $h(\boldsymbol{x})$ becomes

$$h\,(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \underset{C_2}{\overset{C_1}{\substack{> \\ <}}} \ln\frac{P_2}{P_1} \tag{4.131}$$

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 (4.131) becomes

$$h(x(m)) = \frac{\mu_2-\mu_1}{\sigma^2}x(m) + \frac{1}{2}\frac{\mu_2^2-\mu_1^2}{\sigma^2} \underset{C_2}{\overset{C_1}{\gtrless}} 0 \tag{4.132}$$

Hence the rule for signal classification becomes

$$x(m) \underset{C_2}{\overset{C_1}{\lessgtr}} \frac{\mu_1+\mu_2}{2} \tag{4.133}$$

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

$$P\,(\text{Error}|x) = P(C_1)P(x > \text{Thrsh}|x \in C_1) + P(C_2)P(x < \text{Thrsh}|x \in C_2) \tag{4.134}$$

For two Gaussian-distributed classes of scalar-valued signals with pdfs $\mathcal{N}(x(m),\mu_1,\sigma_1^2)$ and $\mathcal{N}(x(m),\mu_2,\sigma_2^2)$, Equation (4.134) becomes

$$\begin{aligned} P(\text{Error}|x) = P(C_1) &\int_{\text{Thrsh}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_i^2}\right) dx \\ + P(C_2) &\int_{-\infty}^{\text{Thrsh}} \frac{1}{\sqrt{2\pi}\sigma_2} \left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right) dx \end{aligned} \tag{4.135}$$

where the parameter Thrsh is the classification threshold.

4.6.3 Bayesian Classification of Discrete-Valued Parameters

Let the set $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_i,\ i = 1,\ldots,M\}$ denote the values that a discrete P-dimensional parameter vector $\boldsymbol{\theta}$ can assume. In general, the observation space $\boldsymbol{Y}$ associated with a discrete parameter space $\boldsymbol{\Theta}$ 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 $\boldsymbol{\theta}_i$, given observation $\boldsymbol{y}$, may be expressed, using Bayes' rule, as

$$P_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}_i|\boldsymbol{y}) = \frac{f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}_i)P_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_i)}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \tag{4.136}$$

For the case when the observation space $\boldsymbol{y}$ is discrete-valued, the probability density functions are replaced by the appropriate probability mass functions. The Bayesian risk in selecting the parameter vector $\boldsymbol{\theta}_i$ given the observation $\boldsymbol{y}$ is defined as

$$\mathcal{R}(\boldsymbol{\theta}_i|y) = \sum_{j=1}^{M} C(\boldsymbol{\theta}_i|\boldsymbol{\theta}_j)P_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}_j|\boldsymbol{y}) \tag{4.137}$$

where $C(\boldsymbol{\theta}_i|\boldsymbol{\theta}_j)$ is the cost of selecting the parameter $\boldsymbol{\theta}_i$ when the true parameter is $\boldsymbol{\theta}_j$. The Bayesian classification Equation (4.137) can be employed to obtain the maximum a posteriori, the maximum likelihood or the minimum mean square error classifiers, as described next.

4.6.4 Maximum A Posteriori Classification

MAP classification corresponds to Bayesian classification with a uniform cost function defined as

$$C(\boldsymbol{\theta}_i|\boldsymbol{\theta}_j) = 1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \tag{4.138}$$

where $\delta(\cdot)$ is the delta function. Substitution of this cost function in the Bayesian risk function yields

$$\begin{aligned} \mathcal{R}_{MAP}(\boldsymbol{\theta}_i|\boldsymbol{y}) &= \sum_{j=1}^{M} [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)] P_{\boldsymbol{\Theta}|y}(\boldsymbol{\theta}_j|\boldsymbol{y}) \\ &= 1 - P_{\boldsymbol{\Theta}|y}(\boldsymbol{\theta}_i|\boldsymbol{y}) \end{aligned} \tag{4.139}$$

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 (4.139) minimisation of the MAP risk function is achieved by maximisation of the posterior pmf:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{MAP}(\boldsymbol{y}) &= \arg\max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}_i|\boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_i) f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}_i) \end{aligned} \tag{4.140}$$

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:

$$\begin{aligned} \mathcal{R}_{ML}(\boldsymbol{\theta}_i|\boldsymbol{y}) &= \sum_{j=1}^{M} [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)] \frac{1}{f_Y(\boldsymbol{y})} f_{Y|\boldsymbol{\Theta}}\left(\boldsymbol{y}|\boldsymbol{\theta}_j\right) P_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_j) \\ &= 1 - \frac{1}{f_Y(\boldsymbol{y})} f_{Y|\theta}(\boldsymbol{y}|\boldsymbol{\theta}_i)\ P_{\boldsymbol{\Theta}} \end{aligned} \tag{4.141}$$

where $P_{\boldsymbol{\Theta}}$ is the uniform pmf of $\boldsymbol{\theta}$. Minimisation of the ML risk function (4.141) is equivalent to maximisation of the likelihood:

$$\hat{\boldsymbol{\theta}}_{ML}(\boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}_i} f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}_i) \tag{4.142}$$

4.6.6 Minimum Mean Square Error Classification

The Bayesian minimum mean square error classification results from minimisation of the risk function

$$\mathcal{R}_{MMSE}(\boldsymbol{\theta}_i|\boldsymbol{y}) = \sum_{j=1}^{M} \left|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\right|^2 P_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}_j|\boldsymbol{y}) \tag{4.143}$$

For the case when $P_{\boldsymbol{\Theta}|Y}(\boldsymbol{\theta}_j|\boldsymbol{y})$ is not available, the MMSE classifier is given by

$$\hat{\boldsymbol{\theta}}_{MMSE}(\boldsymbol{y}) = \arg\min_{\boldsymbol{\theta}_i} |\boldsymbol{\theta}_i - \boldsymbol{\theta}(\boldsymbol{y})|^2 \tag{4.144}$$

where $\boldsymbol{\theta}(\boldsymbol{y})$ is an estimate based on the observation $\boldsymbol{y}$.

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 as $S = \{s_1, s_2, \ldots, s_M\}$, 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

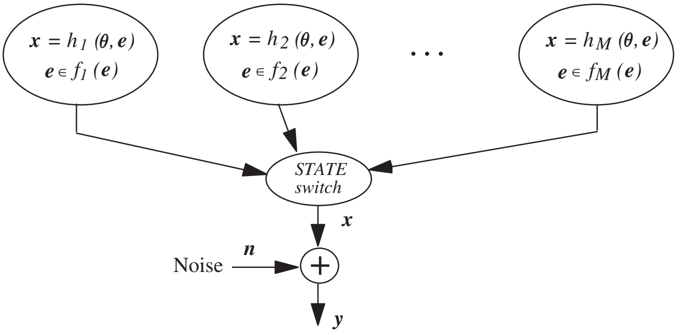
$$\boldsymbol{x}(m) = h_i\left(\boldsymbol{\theta}_i, \boldsymbol{e}(m)\right), \quad i = 1, \ldots, M \tag{4.145}$$

where in each state the signal $x(m)$ is modelled as the output of a state-dependent function $h_i(\cdot)$ with parameter $\boldsymbol{\theta}_i$, input $e(m)$ and an input pdf $f_{Ei}(e(m))$. Each state may be a model of a segment of speech or image. The prior probability of each state is given by

$$P_S(s_i) = \mathcal{E}[N(s_i)] \bigg/ \mathcal{E}\left[\sum_{j=1}^{M} N(s_j)\right] \tag{4.146}$$

where $\mathcal{E}[N(s_i)]$ is the expected number of observation 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

$$f_X\big(\boldsymbol{x}(m)\big) = \sum_{i=1}^{M} P_S(s_i) f_{X|S}(\boldsymbol{x}|s_i) \tag{4.147}$$



$\boldsymbol{x} = h_1(\boldsymbol{\theta}, \boldsymbol{e})$
$\boldsymbol{e} \in f_1(\boldsymbol{e})$
$\boldsymbol{x} = h_2(\boldsymbol{\theta}, \boldsymbol{e})$
$\boldsymbol{e} \in f_2(\boldsymbol{e})$
. . .
$\boldsymbol{x} = h_M(\boldsymbol{\theta}, \boldsymbol{e})$
$\boldsymbol{e} \in f_M(\boldsymbol{e})$
STATE
switch
$\boldsymbol{x}$
$\boldsymbol{n}$
Noise
+
$\boldsymbol{y}$

Figure 4.19 Illustration of a random process generated by a finite state system.

In Figure 4.19, the noisy observation $\boldsymbol{y}(m)$ is the sum of the process output $\boldsymbol{x}(m)$ and an additive noise $\boldsymbol{n}(m)$. From Bayes' rule, the posterior probability of the state s_i given the observation $\boldsymbol{y}(m)$ can be expressed as

$$P_{S|Y}\big(s_i \,|\boldsymbol{y}(m)\big) = \frac{f_{Y|S}\big(\boldsymbol{y}\,(m) \,|\, s_i\big)P_S(s_i)}{\sum\limits_{j=1}^{M} f_{Y|S}\big(\boldsymbol{y}\,(m) \,\big|\, s_j\big)P_S(s_j)} \tag{4.148}$$

In MAP classification, the state with the maximum posterior probability is selected as

$$s_{MAP}\,(\boldsymbol{y}(m)) = \arg\max_{s_i}\, P_{S|Y}\,(s_i\,|\,\boldsymbol{y}(m)\,) \tag{4.149}$$

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

$$R\big(s_i|y(m)\big) = \sum_{j=1}^{M} C(s_i|s_j)P_{S|Y}(s_j|y(m)) \tag{4.150}$$

4.6.8 Bayesian Estimation of the Most Likely State Sequence

Consider the estimation of the most likely state sequence of a finite state process, given a sequence of T observation vectors. A state sequence $\boldsymbol{s}$, of length T, is itself a random integer-valued vector process with M^T possible values where M is the number of states. From Bayes' rule, the posterior pmf of a state sequence $\boldsymbol{s}$, given an observation sequence $\boldsymbol{Y}$, can be expressed as

$$P_{S|Y}(s_{i_0},\,\dots\,,s_{i_{T-1}}\,|\boldsymbol{y}_0,\,\dots\,,\boldsymbol{y}_{T-1}\,) = \frac{f_{Y|S}(\boldsymbol{y}_0,\,\dots\,,\boldsymbol{y}_{T-1}\,|s_{i_0},\,\dots\,,s_{i_{T-1}}\,)P_S(s_{i_0},\,\dots\,,s_{i_{T-1}})}{f_Y(\boldsymbol{y}_0,\,\dots\,,\boldsymbol{y}_{T-1})} \tag{4.151}$$

where $P_S(s)$ is the pmf of the state sequence $\boldsymbol{s}$, and for a given observation sequence, the denominator is a constant. The Bayesian risk in selecting a state sequence s_i is expressed as

$$\mathcal{R}(s_i|\boldsymbol{Y}) = \sum_{j=1}^{M^T} C(s_i|s_j)P_{S|Y}(s_j|\boldsymbol{Y}) \tag{4.152}$$

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

$$P_{S|Y}(s_{i_0},\,\dots\,,s_{i_{T-1}}\,|\boldsymbol{y}_0,\,\dots\,,\boldsymbol{y}_{T-1}\,) = \prod_{k=0}^{T-1} f_{Y|S}(\boldsymbol{y}_k\,|s_{i_k}\,)\,P_S(s_{i_k}) \tag{4.153}$$

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 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

$$P_{S|Y}(s_{i_0},\,\dots\,,s_{i_{T-1}}\,|\boldsymbol{y}_0,\,\dots\,,\boldsymbol{y}_{T-1}) = \prod_{k=0}^{T-1} a_{i_{k-1}i_k} f_{S|Y}(s_{i_k}\,|\boldsymbol{y}_k) \tag{4.154}$$

where is the probability that the process moves from state to state. Finite state random processes and computationally efficient methods of state sequence estimation are described in detail in Chapter 5.

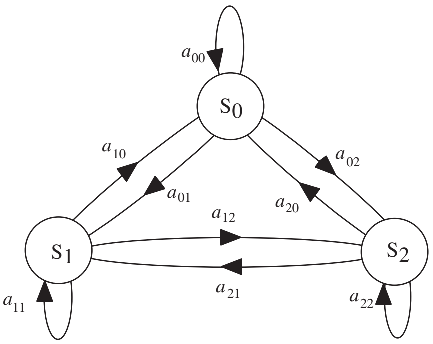


Figure 4.20 A three-state Markov process.

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 quantisation is 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 $\boldsymbol{X}$, is partitioned into K clusters or regions $[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K]$ and each cluster $\boldsymbol{X}_i$ is represented by a cluster centroid $\boldsymbol{c}_i$. The set of centroid vectors $[\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k]$ forms a VQ codebook model for the process X.

The VQ codebook can then be used to classify an unlabelled vector $\boldsymbol{x}$ with the nearest centroid. The codebook is searched to find the centroid vector with the minimum distance from $\boldsymbol{x}$, then $\boldsymbol{x}$ is labelled with the index of the minimum distance centroid as

$$\text{Label}(\boldsymbol{x}) = \arg\min_i \ d(\boldsymbol{x}\boldsymbol{,}\boldsymbol{c}_i) \tag{4.155}$$

where $d(\boldsymbol{x}, \boldsymbol{c}_i)$ is a measure of distance between the vectors $\boldsymbol{x}$ and $\boldsymbol{x}_i$. The most commonly used distance measure is the mean squared distance.

4.7.2 Vector Quantisation using Gaussian Models of Clusters

In vector quantisation, instead of using only the centre or mean value of each cluster, a Gaussian pdf model of each cluster comprising the mean 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 $\boldsymbol{X}$, is partitioned into K clusters or regions $[\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{x}_K]$ and each cluster $\boldsymbol{X}_{\text{i}}$ is represented by a cluster centroid vector $\boldsymbol{c}_i$, the cluster covariance matrix $\boldsymbol{\Sigma}_i$ and the cluster probability p_i as $[\mathcal{N}(\boldsymbol{c}_i, \boldsymbol{\Sigma}_i), p_i]$. The set of VQ pdfs

$\{[\mathcal{N}(\boldsymbol{c}_1, \boldsymbol{\Sigma}_1), p_1], [\mathcal{N}(\boldsymbol{c}_2, \boldsymbol{\Sigma}_2), p_2], \ldots, [\mathcal{N}(\boldsymbol{c}_K, \boldsymbol{\Sigma}_K), p_K]\}$ forms a VQ codebook model for the process $\boldsymbol{X}$. The VQ codebook can then be used to classify an unlabelled vector $\boldsymbol{x}$ with the nearest pdf. The codebook is searched to find the VQ pdf with the maximum probability of membership for $\boldsymbol{x}$, then $\boldsymbol{x}$ is labelled with the index of the nearest pdf as

$$\text{Label}(\boldsymbol{x}) = \arg\min_{i}\ p_i\ \mathcal{N}(\boldsymbol{x},\boldsymbol{c}_i, \boldsymbol{\Sigma}_i) \tag{4.156}$$

where the weighted Gaussian pdf distance $p_i\mathcal{N}(\boldsymbol{x},\boldsymbol{c}_i, \boldsymbol{\Sigma}_i)$, is a measure of membership of the input vector $\boldsymbol{x}$ and the VQ class i.

4.7.3 Design of a Vector Quantiser: K-Means Clustering

The K-means algorithm, illustrated in Figures 4.21 and 4.22, is an iterative method for the design of a VQ codebook. Each iteration consists of two basic steps:

(1) partition the training signal space into K regions or clusters; and
(2) compute the centroid of each region.

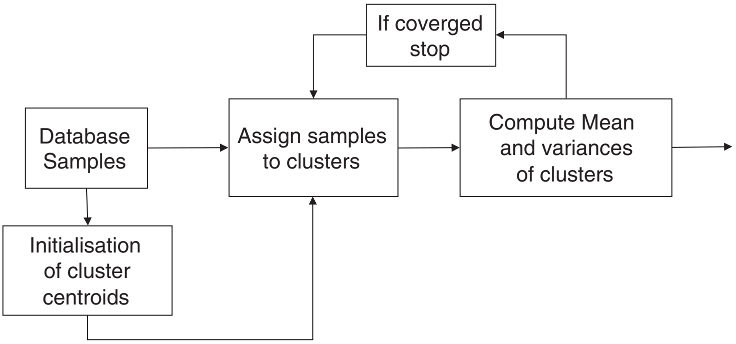
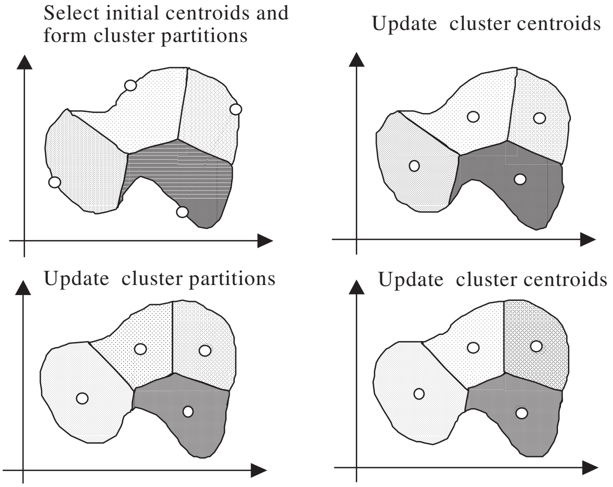

If coverged stop
Database Samples
Assign samples to clusters
Compute Mean and variances of clusters
Initialisation of cluster centroids

Figure 4.21 Illustration of the K-means algorithm.


Select initial centroids and form cluster partitions

Update cluster centroids

Update cluster partitions

Update cluster centroids

Figure 4.22 Illustration of the working of the K-means clustering method.

The steps in the K-means method are as follows:

Step 1: Initialisation Use a suitable method to choose a set of K initial centroids $[\boldsymbol{c}_i]$ for $m = 1, 2, \ldots$

Step 2: Classification Classify the training vectors $\{\boldsymbol{x}\}$ into K clusters $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K\}$ using the so-called nearest-neighbour rule, Equation (4.154).

Step 3: Centroid computation Use the vectors $[\boldsymbol{x}_i]$ associated with the i^{th} cluster to compute an updated cluster centroid $\boldsymbol{c}_i$, and calculate the cluster distortion defined as

$$D_i(m) = \frac{1}{N_i} \sum_{j=1}^{N_i} d(\boldsymbol{x}_i(j), \boldsymbol{c}_i(m)) \tag{4.157}$$

where it is assumed that a set of N_i vectors $[\boldsymbol{x}_i(j)\ j = 0, \ \ldots, \ N_i]$ are associated with cluster i. The total distortion is given by

$$D(m) = \sum_{i=1}^{K} D_i(m) \tag{4.158}$$

Step 4: Convergence test:

```
    if
        $D(m-1) - D(m) >$ Threshold, stop,
    else
  go to 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 non-stationary processes studied in Chapter 5 and the blind equalisation of distorted signals studied in Chapter 16.

We considered a number of examples of the estimation of a signal observed in noise and derived 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 telecommunication 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, which 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 estimate-maximise 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 problem of vector quantisation, the popular K-means clustering methods and the modelling of a data space with a mixture Gaussian process. We 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. Stat*. **41**: 164–171.

Bayes T. (1763) An Essay Towards Solving a Problem in the Doctrine of Changes, *Phil. Trans. Royal Society of London*, **53**: 370–418, (reprinted in 1958 in *Biometrika*, **45**: 293–315).

Bezdek J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Chou P., Lookabaugh T. and Gray R. (1989) Entropy-Constrained Vector Quantisation. *IEEE Trans*. Acoustics, Speech and Signal Processing, ASSP-**37**: 31–42.

Cramer H. (1974) *Mathematical Methods of Statistics*. Princeton University Press.

Dempster A.P., Laird N.M. and Rubin D.B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *J.R. Stat. 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*. John Wiley & Sons, Inc, New York.

Feder M. and Weinstein E. (1988) Parameter Estimation of Superimposed Signals using the EM algorithm. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-36**(4): 477, 489.

Fisher R.A. (1922) On the Mathematical Foundations of the Theoretical Statistics. *Phil Trans. Royal. Soc. London*, **222**: 309–368.

Gersho A. (1982) On the Structure of Vector Quantisers. IEEE Trans. Information Theory, **IT-28**: 157–166.

Gray R.M. (1984) Vector Quantisation. *IEEE ASSP Magazine*: 4–29.

Gray R.M. and Karnin E.D (1982), Multiple Local Optima in Vector Quantisers. *IEEE Trans*. Information Theory, IT-**28**: 256–261.

Holland J.D. (1962) The Reverend Thomas Bayes, F.R.S. (1702–61) *Journal of the Royal Statistical Society. Series A (General)*, **125**(3): 451–461.

Jeffrey H. (1961) *Scientific Inference*, 3rd edn. Cambridge University Press.

Larson H.J. and Bruno O.S. (1979) *Probabilistic Models in Engineering Sciences*. I and II. John Wiley & Sons, Inc, New York.

Linde Y., Buzo A. and Gray R.M. (1980) An Algorithm for Vector Quantiser Design. *IEEE Trans. Comm*. **COM-28**: 84–95.

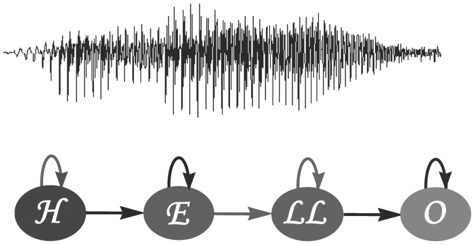
Makhoul J., Roucos S., and Gish H. (1985) Vector Quantisation in Speech Coding. *Proc. IEEE*, **73**: 1551–1588.

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 Review*, **26**: 195–239.

Scharf L.L. (1991) *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison Wesley, Reading, MA.



5

Hidden Markov Models

Hidden Markov models (HMMs) are used for statistical modelling of non-stationary signal processes such as speech signals, image sequences time-varying noise, bio-signals and DNA sequences. A Markov process, developed by Andrei Markov, is a process whose state or value at time t depends on its previous state or values at time $t-1$ and is independent of the history of the process before time $t-1$. 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 non-stationary random process with a Markovian chain of state-dependent stationary sub-processes. 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 non-stationary 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 HMMs for the enhancement of noisy signals are considered.

5.1 Statistical Models for Non-Stationary Processes

A non-stationary process can be defined as one whose statistical parameters, such as its moments, vary over time. Most ‘naturally generated’ signals, such as audio signals, video signals, biomedical signals and seismic signals, are non-stationary, 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 non-stationary 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, non-stationary processes can be classified into one of two broad categories:

(1) *continuously variable state* processes,
(2) *finite state* processes.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

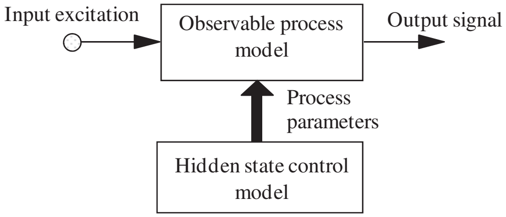


Figure 5.1 Illustration of a two-layered model of a non-stationary signal process where a hidden process controls the state of the observable process.

A continuously variable state process is defined as one whose underlying statistics vary continuously with time. Examples of this class of random processes are most audio signals, such as speech, the power and spectral composition of which vary continuously with time. A finite state process is one whose statistical characteristics can *switch* between a finite number of stationary or non-stationary states. For example, impulsive noise is a binary-state process and across different phonetic segments speech is a finite state process. Note that a continuously variable process can be approximately expressed through 'quantisation' of its time-varying statistical variations in terms of a chain of finite state processes.

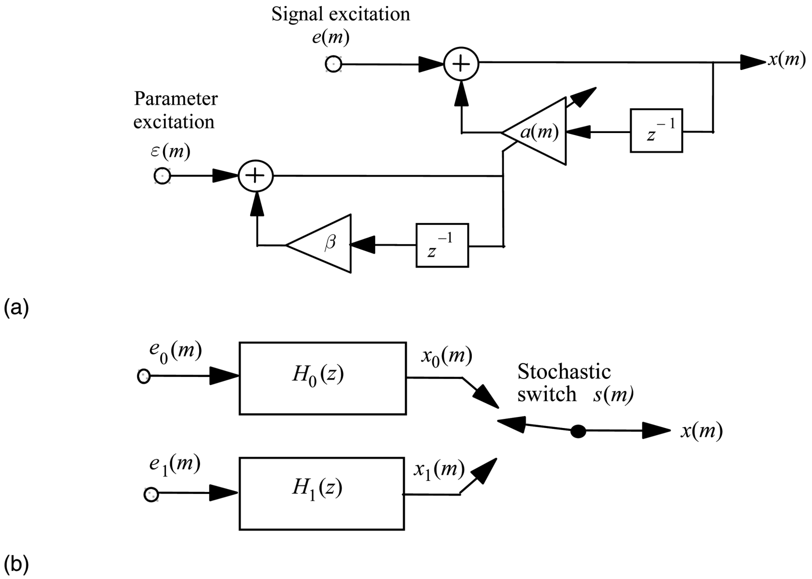


Figure 5.2 (a) A continuously variable state AR process. (b) A binary-state AR process.

Figure 5.2(a) illustrates a non-stationary first-order autoregressive (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 non-stationary AR model. For this model, the observation signal equation and the hidden parameter state equation can be expressed as

$$x(m) = a(m)x(m-1) + e(m) \quad \text{Observation equation} \tag{5.1}$$

$$a(m) = \beta a(m-1) + \varepsilon(m) \quad \text{Hidden state equation} \tag{5.2}$$

where $a(m)$ is the time-varying coefficient of the observable AR process and β is the coefficient of the hidden state-control process.

A simple example of a finite-state non-stationary model is the binary-state autoregressive (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

This section provides a detailed introduction to the definition, structure, parameters and workings of hidden Markov models.

5.2.1 Comparison of Markov and Hidden Markov Models

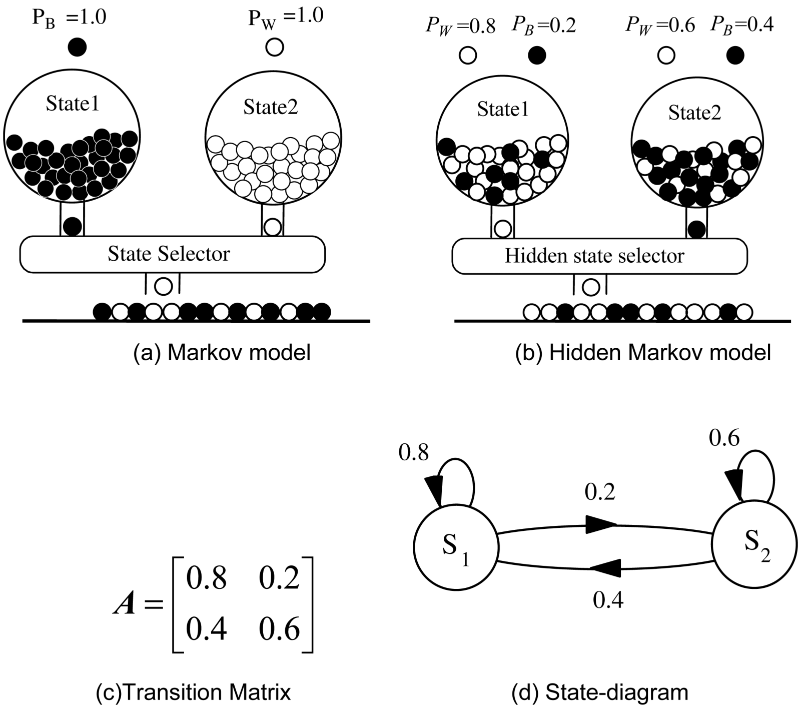
A Markov process is defined as stochastic random process whose probability of being in a given state at time m depends on the previous state of the process at time $m - 1$ and is independent of the states of the process before $m - 1$.

5.2.1.1 Observable-State Markov Process

Consider a Markov process whose state sequence is observable from its output sequence. A simple example of a two-state Markov process is illustrated in Figure 5.3(a) which shows two containers (states): in state 1 the process always (with a probability of $P_{\mathrm{B}} = 1$) outputs black balls and in state 2 the process always (with a probability of $P_{\mathrm{W}} = 1$) outputs white balls. Now assume that at successive time intervals a random selection process selects one of the two containers to release a ball. Assume also that the state selection process is Markovian 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 of moving from state 2 to state 1 is $p(s_{1,m}|s_{2,m-1}) = 0.4$. Note that, since each container produces a unique output, the Markovian process equally models the container selection process and the container output process.

5.2.1.2 Hidden-State Markov Process

A hidden Markov model (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 as P_{B} and P_{W} respectively, are as shown in the paragraph above. 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 the black and the white balls in each container remains unaffected. Each container can be considered as an underlying state of the output process. Now for an example assume that the hidden container-selection process is governed by the following rule: at any time, if the output from the



(a) Markov model

(b) Hidden Markov model

$$\boldsymbol{A} = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}$$

(c)Transition Matrix

(d) State-diagram

Figure 5.3 (a) A Markov model, each state is identified by the output, (b) a hidden Markov model (HMM), 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.

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.

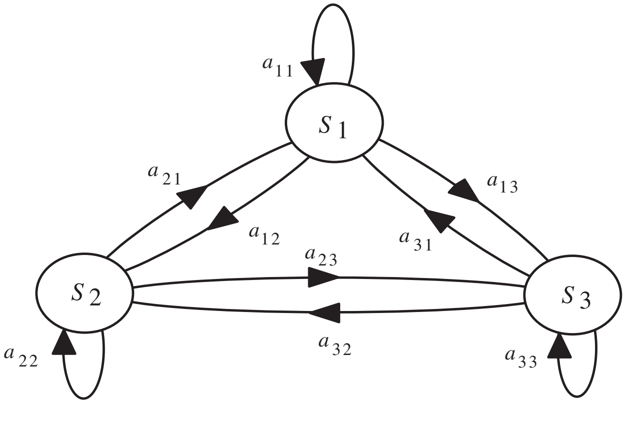


Figure 5.4 A three-state ergodic HMM structure.

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 quasi-stationary, elementary sub-processes. The 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 term 'ergodic' implies that there are no structural constraints for connecting any one 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 the spoken letter '*C*', phonetically transcribed as '*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.

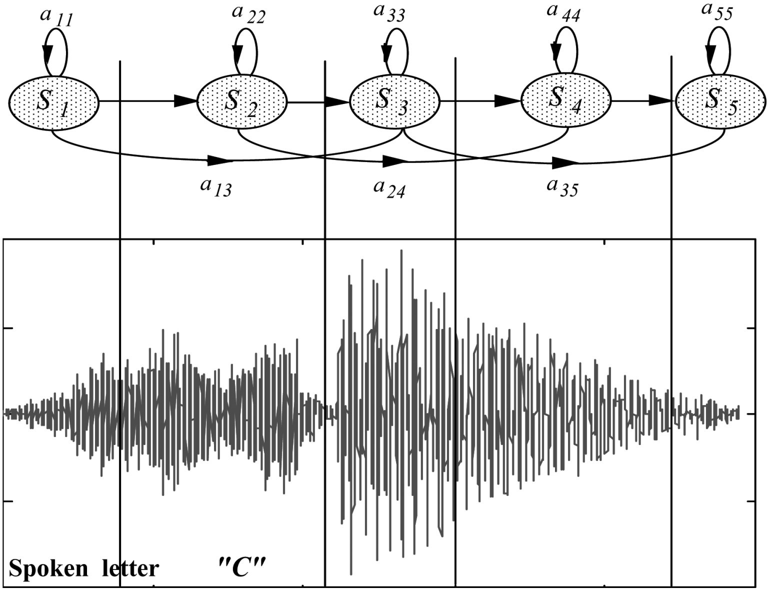


Figure 5.5 A five-state left–right HMM speech model.

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 $\boldsymbol{s}$ of a model $\mathcal{M}$, given a sequence of observation vectors $\boldsymbol{X} = [\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(T-1)]$, can be expressed using Bayes' rule as the product of the *prior* probability of the state sequence $\boldsymbol{s}$ and the *likelihood* of the observation sequence $\boldsymbol{X}$ as

$$P_{S|X,\mathcal{M}}(\boldsymbol{s}|\boldsymbol{X},\mathcal{M}) = \frac{1}{f_X(\boldsymbol{X})} \underbrace{P_{S|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})}_{\text{State prior}} \underbrace{f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})}_{\text{Observation likelihood}} \tag{5.4}$$

where the likelihood of the observation sequence $\boldsymbol{X}$ is modelled by a probability density function $f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})$.

The posterior probability that an observation signal sequence $\boldsymbol{X}$ was generated by the model $\mathcal{M}$ is summed over all likely state sequences, and may also be weighted by the model prior $P_{\mathcal{M}}(\mathcal{M})$.

$$P_{\mathcal{M}|X}(\mathcal{M}|\boldsymbol{X}) = \frac{1}{f_X(\boldsymbol{X})} \underbrace{P_{\mathcal{M}}(\mathcal{M})}_{\text{Model prior}} \sum_{s} \underbrace{P_{S|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})}_{\text{State prior}} \underbrace{f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})}_{\text{Observation likelihood}} \tag{5.5}$$

The Markovian state transition prior can be used to model the time variations and the sequential dependence of most non-stationary 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 to 5.

State transition-probability matrix $\boldsymbol{A} = \{a_{ij}, i,j = 1, \ldots, N\}$. 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 $\boldsymbol{A}$ is upper-triangular.

State observation vectors $\{\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \ldots, \boldsymbol{\mu}_{iM}, i = 1, \ldots, N\}$. For each state a set of M prototype vectors model 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 $\boldsymbol{P} = \{P_{ij}(\cdot);\ i = 1, \ldots, N,\ j = 1, \ldots, M\}$, or it may be a continuous (usually Gaussian) pdf model $\boldsymbol{F} = \{f_{ij}(\cdot);\ i = 1, \ldots, N, j = 1, \ldots, M\}$.

Initial state probability vector $\boldsymbol{\pi} = [\pi_1,\ \pi_2, \ldots,\ \pi_N]$.

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 modelling 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 $[\boldsymbol{\mu}_{i1},\ \ldots,\ \boldsymbol{\mu}_{iM}]$ with a pmf $[P_{i1},\ \ldots,\ P_{iM}]$. 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 $\{\boldsymbol{\mu}_{ik},\ P_{ik};\ i = 1, \ldots,\ N,\ k = 1,\ \ldots,\ M\}$. In effect, this results in a discrete observation HMM for a continuous-valued process.

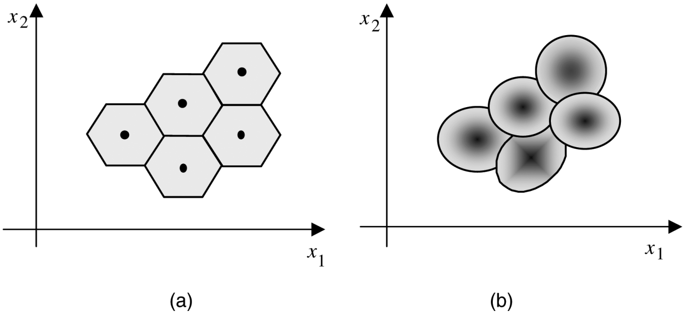


Figure 5.6 Modelling a random signal space using (a) a discrete-valued pmf and (b) a continuous-valued mixture Gaussian density.

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

$$f_{X|S}\ (\boldsymbol{x}\,|s = i\,) = \sum_{k=1}^{M} P_{ik}\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \tag{5.6}$$

where $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{ik},\ \boldsymbol{\Sigma}_{ik})$ 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 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.

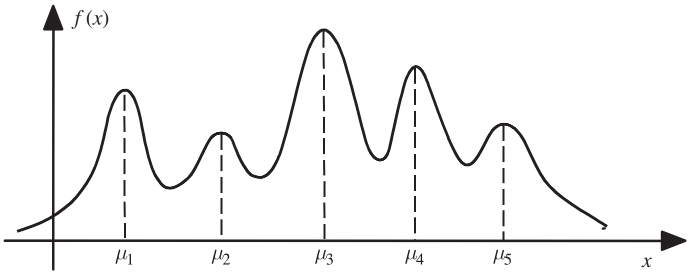


Figure 5.7 A mixture Gaussian probability density function.

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

$$\begin{aligned} &Prob\big(s(t)=j\,|s(t-1)=i, s(t-2)=k, \ldots, s(t-N)=l\,\big) \\ &\quad = Prob\big(s(t)=j\,|s(t-1)=i\big)=a_{ij} \end{aligned} \tag{5.7}$$

where $s(t)$ denotes the state of the 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

$$P_i\,(d)=a_{ii}^{d-1}\,(1-a_{ii}) \tag{5.8}$$

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. Figures 5.8(a) and 5.8(b) illustrate 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 $\boldsymbol{s}=[S_1, S_1, S_2, S_2, S_3, S_3, S_4]$ is $P(\boldsymbol{s})=\pi_1 a_{11} a_{12} a_{22} a_{23} a_{33} a_{34}$. 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.

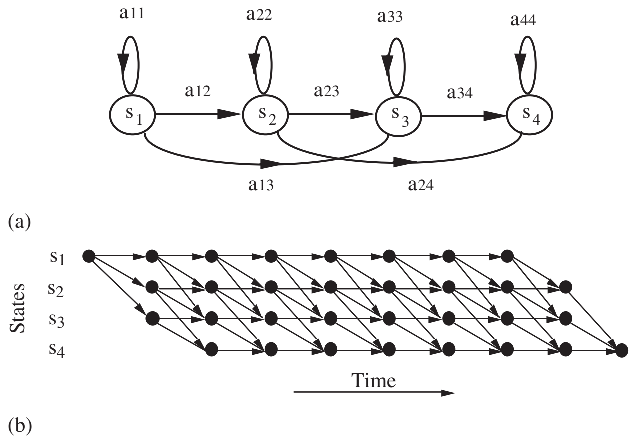


Figure 5.8 (a) A four-state left–right HMM, and (b) its state–time trellis diagram.

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 $[\boldsymbol{X}] = [\boldsymbol{X_k};\ k = 0,\ \ldots,\ L-1]$, with each sequence $\boldsymbol{X_k} = [\boldsymbol{x}(t);\ t = 0,\ \ldots,\ T_k - 1]$ 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 $P_{\mathcal{M}|X}(\mathcal{M}|\boldsymbol{X})$ of the model $\mathcal{M}$ given the training data $\boldsymbol{X}$. The training process aims to maximise the posterior probability of the model $\mathcal{M}$ and the training data $[\boldsymbol{X}]$, expressed using Bayes' rule as

$$P_{\mathcal{M}|X}(\mathcal{M}|\boldsymbol{X}) = \frac{1}{f_X(\boldsymbol{X})} f_{X|\mathcal{M}}(\boldsymbol{X}\,|\mathcal{M})\, P_{\mathcal{M}}(\mathcal{M}) \tag{5.10}$$

where the denominator $f_X(\boldsymbol{X})$ 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 $[\boldsymbol{X}]$ and a given model $\mathcal{M}$, maximising Equation (5.10) is equivalent to maximising the likelihood function $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$. The likelihood of an observation vector sequence $\boldsymbol{X}$ given a model $\mathcal{M}$ can be expressed as

$$f_{X|\mathcal{M}}(\boldsymbol{X}\,|\mathcal{M}) = \sum_{s} f_{X|S,\mathcal{M}}(\boldsymbol{X}\,|\boldsymbol{s},\mathcal{M})\, P_{s|\mathcal{M}}(\boldsymbol{s}\,|\mathcal{M}) \tag{5.11}$$

Where $f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\ \mathcal{M})$ the pdf of the signal sequence $\boldsymbol{X}$ along the state sequence $\mathbf{s} = [s(0),\ s(1), \ldots,\ s(T-1)]$ of the model $\mathcal{M}$, is given by

$$f_{X|S,\mathcal{M}}\big(\boldsymbol{X}\,|\boldsymbol{s},\mathcal{M}\big) = f_{X|S}\big(\boldsymbol{x}(0)\,|s(0)\,\big) f_{X|S}\big(\boldsymbol{x}(1)\,|s(1)\,\big) \cdots f_{X|S}\big(\boldsymbol{x}(T-1)\,|s(T-1)\,\big) \tag{5.12}$$

where $s(t)$, the state at time t, can be one of N states, and $f_{X|S}(\boldsymbol{X}|s(t))$, a shorthand for $f_{X|S,\mathcal{M}}(\boldsymbol{X}|s(t),\mathcal{M})$, is the pdf of $\boldsymbol{x}(t)$ given the state $s(t)$ of the model $\mathcal{M}$. The Markovian probability of the state sequence $\boldsymbol{s}$ is given by

$$P_{S|\mathcal{M}}(\boldsymbol{s}\,|\mathcal{M}) = \pi_{s(0)} a_{s(0)s(1)} a_{s(1)s(2)} \ldots a_{s(T-2)s(T-1)} \tag{5.13}$$

Substituting Equations (5.12) and (5.13) in Equation (5.11) yields

$$\begin{aligned} f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}) &= \sum_{s} f_{X|S,\mathcal{M}}(X|s,\mathcal{M})\, P_{s|\mathcal{M}}(s|\mathcal{M}) \\ &= \sum_{s} \pi_{s(0)} f_{X|S}\big(\boldsymbol{x}(0)\,|s(0)\big) a_{s(0)s(1)} f_{X|S}\big(\boldsymbol{x}(1)\,|s(1)\big) \\ &\quad \cdots a_{s(T-2)s(T-1)} f_{X|S}\big(x(T-1)\,|s(T-1)\big) \end{aligned} \tag{5.14}$$

where the summation is taken over all state sequences $\boldsymbol{s}$. 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 non-trivial 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 $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$ 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 $\boldsymbol{X}=[\boldsymbol{x}(0),\ \boldsymbol{x}(1),\ \ldots,\ \boldsymbol{x}(t)]$ and the state i at time t, of the model $\mathcal{M}$:

$$\alpha_t(i) = f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(t),s(t)=i\,|\mathcal{M}\big) \tag{5.15}$$

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 $t-1$, $\alpha_{t-1}(i)$:

$$\begin{aligned} \alpha_t(i) &= f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(t),s(t)=i\,|\mathcal{M}\big) \\ &= \left(\sum_{j=1}^{N} f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(t-1),s(t-1)=j\,|\mathcal{M}\big) a_{ji}\right) \\ &\quad f_{X|S,\mathcal{M}}\big(\boldsymbol{x}(t)\,|s(t)=i,\mathcal{M}\big) \\ &= \sum_{j=1}^{N} \big(\alpha_{t-1}(j) a_{ji}\big) f_{X|S,\mathcal{M}}\big(\boldsymbol{x}(t)\,|s(t)=i,\mathcal{M}\big) \end{aligned} \tag{5.16}$$

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 $\boldsymbol{X}=[\boldsymbol{x}(0),\ \boldsymbol{x}(1),\ \ldots,\ \boldsymbol{x}(T-1)]$ given a model $\mathcal{M}$ can be expressed in terms of the forward probabilities as

$$\begin{aligned} f_{X|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(T-1)\,|\mathcal{M}\big) &= \sum_{i=1}^{N} f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(T-1),s(T-1)=i\,|\mathcal{M}\big) \\ &= \sum_{i=1}^{N} \alpha_{T-1}(i) \end{aligned} \tag{5.17}$$

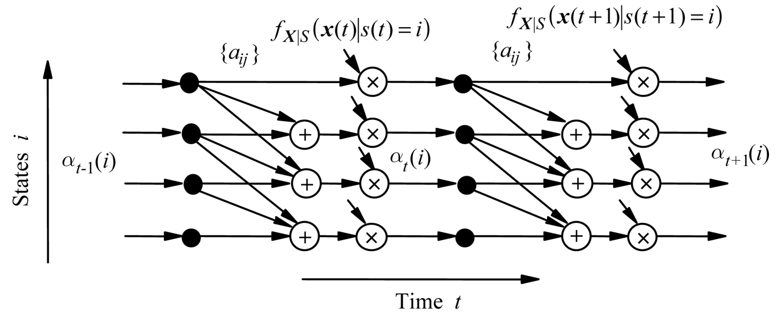


Figure 5.9 A network for computation of forward probabilities for a left–right HMM.

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 $[\boldsymbol{x}(t+1), \boldsymbol{x}(\text{t+2}), \ldots, \boldsymbol{x}(T-1)]$ as

$$\begin{aligned}
\beta_t(i) &= f_{X,S|\mathcal{M}}\left(s(t)=i, \boldsymbol{x}(t+1), \boldsymbol{x}(t+2), \ldots, \boldsymbol{x}(T-1)\,|\mathcal{M}\right) \\
&= \sum_{j=1}^{N} a_{ij} f_{X,S|\mathcal{M}}\left(s(t+1)=j, \boldsymbol{x}(t+2), \boldsymbol{x}(t+3), \ldots, \boldsymbol{x}(T-1)\,|\mathcal{M}\right) \\
&\quad \times\ f_{X|S,\mathcal{M}}\left(x(t+1)\,|s(t+1)=j, \mathcal{M}\right) \\
&= \sum_{j=1}^{N} a_{ij}\beta_{t+1}(j) f_{X|S,\mathcal{M}}\left(\boldsymbol{x}(t+1)\,|s(t+1)=j, \mathcal{M}\right)
\end{aligned} \tag{5.18}$$

In the next section, forward and backward probabilities are used to develop a method for training HMM parameters.

5.3.2 *Baum–Welch Model Re-estimation*

The HMM training problem is the estimation of the model parameters $\mathcal{M} = (\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{F})$ for a given data set $\boldsymbol{X}$. These parameters are the initial state probabilities $\boldsymbol{\pi}$, the state transition probability matrix $\boldsymbol{A}$ and the continuous (or discrete) density state observation pdfs. The HMM parameters are estimated from a set of training examples $\{\boldsymbol{X} = [\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]\}$, with the objective of maximising $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$, 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 $\boldsymbol{X}$, can be expressed as

$$\begin{aligned}
\gamma_t(i,j) &= P_{S|X,\mathcal{M}}\left(s(t)=i, s(t+1)=j\,|\boldsymbol{X}, \mathcal{M}\right) \\
&= \frac{f_{S,X|\mathcal{M}}\left(s(t)=i, s(t+1)=j, \boldsymbol{X}\,|\mathcal{M}\right)}{f_{X|\mathcal{M}}(\boldsymbol{X}\,|\mathcal{M})} \\
&= \frac{\alpha_t(i) a_{ij} f_{X|S,\mathcal{M}}\left(\boldsymbol{x}(t+1)\,|s(t+1)=j, \mathcal{M}\right)\beta_{t+1}(j)}{\sum\limits_{i=1}^{N} \alpha_{T-1}(i)}
\end{aligned} \tag{5.19}$$

where $f_{S,X}|\boldsymbol{M}\big(s(t)=i, s(t+1)=j, \boldsymbol{X}|\mathcal{M}\big)$ is the joint pdf of the states $s(t)$ and $s(t+1)$ and the observation sequence $\boldsymbol{X}$, and $f_{X|S}\,(\boldsymbol{x}(t+1)|s(t+1)=i)$ 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 with the discrete state observation pmf $P_{X|S}\,(\boldsymbol{x}(t+1)|s(t+1)=i)$. The posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is

$$\begin{aligned}\gamma_t(i) &= P_{S|X,\mathcal{M}}\big(s(t)=i\,|\boldsymbol{X},\mathcal{M}\big)\\ &= \frac{f_{S,X|\mathcal{M}}\big(s(t)=i,\boldsymbol{X}\,|\mathcal{M}\big)}{f_{X|\mathcal{M}}\big(\boldsymbol{X}\,|\mathcal{M}\big)}\\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum\limits_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.20}$$

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

$$\bar{a}_{ij} = \frac{\sum\limits_{t=0}^{T-2}\gamma_t(i,j)}{\sum\limits_{t=0}^{T-2}\gamma_t(i)} \tag{5.22}$$

Note that for an observation sequence $[\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]$ 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 $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \ldots, \boldsymbol{\mu}_{iM}]$ 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 $f_{X|S}\,(\boldsymbol{x}(t)|s(t)=i)$ are substituted with probability mass functions $P_{X|S}\,(\boldsymbol{x}(t)|s(t)=i)$ defined as

$$P_{X|S}\big(\boldsymbol{x}(t)\,|s(t)=i\big) = P_{X|S}\big(Q[\boldsymbol{x}(t)]\,|s(t)=i\big) \tag{5.24}$$

where the function vector $Q[\boldsymbol{x}(t)]$ quantises the observation vector $\boldsymbol{x}(t)$ to the nearest discrete vector in the set $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \ldots, \boldsymbol{\mu}_{iM}]$. 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:

$$\begin{aligned}\overline{P}_{ik}(\boldsymbol{\mu}_{ik}) &= \frac{\text{expected number of times in state } i \text{ and observing } \boldsymbol{\mu}_{ik}}{\text{expected number of times in state } i} \\ &= \frac{\sum\limits_{t \in x(t) \to \boldsymbol{\mu}_{ik}}^{T-1} \gamma_t(i)}{\sum\limits_{t=0}^{T-1} \gamma_t(i)}\end{aligned} \tag{5.25}$$

In Equation (5.25) the summation in the numerator is taken over those time instants $t \in x(t) \to \boldsymbol{\mu}_{ik}$ 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 $\boldsymbol{X}$ consisting of a sufficient number of independent realisations of the process to be modelled. Assume that the training data set consists of L realisations $\boldsymbol{X} = [\boldsymbol{X}(0), \boldsymbol{X}(1), \ldots, \boldsymbol{X}(L-1)]$, where $\boldsymbol{X}(k) = [\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(T_k - 1)]$. The re-estimation formula can be averaged over the entire data set as

$$\hat{\pi}_i = \frac{1}{L} \sum_{l=0}^{L-1} \gamma_0^l(i) \tag{5.26}$$

$$\hat{a}_{ij} = \frac{\sum\limits_{l=0}^{L-1} \sum\limits_{t=0}^{T_l-2} \gamma_t^l(i,j)}{\sum\limits_{l=0}^{L-1} \sum\limits_{t=0}^{T_l-2} \gamma_t^l(i)} \tag{5.27}$$

and

$$\hat{P}_i(\boldsymbol{\mu}_{ik}) = \frac{\sum\limits_{l=0}^{L-1} \sum\limits_{t \in x(t) \to \boldsymbol{\mu}_{ik}}^{T_l-1} \gamma_t^l(i)}{\sum\limits_{l=0}^{L-1} \sum\limits_{t=0}^{T_l-1} \gamma_t^l(i)} \tag{5.28}$$

In Equation (5.28) the inner summation in the numerator is taken over those time instants $t \in x(t) \to \boldsymbol{\mu}_{ik}$ where the k^{th} symbol $\boldsymbol{\mu}_{ik}$ is observed in the state i. 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 probability density functions (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

$$f_{X|S}\left(\boldsymbol{x}(t)\,|s(t)=i\right) = \frac{1}{(2\pi)^{P/2}\,|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left\{-[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]^{\mathrm{T}}\,\boldsymbol{\Sigma}_i^{-1}\,[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]\right\} \tag{5.29}$$

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

$$\overline{\boldsymbol{\mu}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)\boldsymbol{x}(t)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.30}$$

Similarly, the covariance matrix is estimated as

$$\overline{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)\big(\boldsymbol{x}(t)-\overline{\boldsymbol{\mu}}_i\big)\big(\boldsymbol{x}(t)-\overline{\boldsymbol{\mu}}_i\big)^{\mathrm{T}}}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.31}$$

The proof that the Baum–Welch re-estimation algorithm leads to maximisation of the likelihood function $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$ 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 Chapter 4. In HMMs with Gaussian mixture pdf for state observation model, the signal space associated with the i^{th} state is modelled with a mixture of M Gaussian densities as

$$f_{X|S}\big(\boldsymbol{x}(t)\,|s(t)=i\big) = \sum_{k=1}^{M} P_{ik}\mathcal{N}\big(\boldsymbol{x}(t), \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}\big) \tag{5.32}$$

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 $\boldsymbol{X} = [\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]$, can be expressed as

$$\begin{aligned}\gamma_t(i,j) &= P_{S|X,\mathcal{M}}\big(s(t)=i, s(t+1)=j|\boldsymbol{X}, \mathcal{M}\big)\\ &= \frac{\alpha_t(i)a_{ij}\left[\sum_{k=1}^{M} P_{jk}\mathcal{N}\big(\boldsymbol{x}(t+1), \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}\big)\right]\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_{T-1}(i)}\end{aligned} \tag{5.33}$$

and the posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is given by

$$\begin{aligned}\gamma_t(i) &= P_{S|X,\mathcal{M}}\big(s(t)=i\,|\boldsymbol{X}, \mathcal{M}\big)\\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.34}$$

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

$$\begin{aligned}\zeta_t(i,k) &= P_{S,K|X,\mathcal{M}}\big(s(t)=i, m(t)=k\,|\boldsymbol{X},\mathcal{M}\big)\\ &= \frac{\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji}P_{ik}\mathcal{N}\big(\boldsymbol{x}(t),\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik}\big)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.35}$$

where $m(t)$ is the Gaussian mixture component at time t. Equations (5.33) to (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

$$\begin{aligned}\overline{P}_{ik} &= \frac{\text{expected number of times in state } i \text{ and observing mixture } k}{\text{expected number of times in state } i}\\ &= \frac{\sum_{t=0}^{T-1}\xi_t(i,k)}{\sum_{t=0}^{T-1}\gamma_t(i)}\end{aligned} \tag{5.36}$$

and

$$\overline{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=0}^{T-1}\xi_t(i,k)\boldsymbol{x}(t)}{\sum_{t=0}^{T-1}\xi_t(i,k)} \tag{5.37}$$

Similarly the covariance matrix is estimated as

$$\overline{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{t=0}^{T-1}\xi_t(i,k)\,[\boldsymbol{x}(t)-\overline{\boldsymbol{\mu}}_{ik}]\,[\boldsymbol{x}(t)-\overline{\boldsymbol{\mu}}_{ik}]^{\mathrm{T}}}{\sum_{t=0}^{T-1}\xi_t(i,k)} \tag{5.38}$$

5.4 Decoding Signals Using Hidden Markov Models

Hidden Markov models are used in applications such as speech recognition, image recognition and signal restoration, and for decoding 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.

Consider the decoding of an unlabelled sequence of T signal vectors $\boldsymbol{X}=[\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\ \boldsymbol{x}(T-1)]$ given a set of V candidate HMMs $[\mathcal{M}_1,\ldots,\ \mathcal{M}_V]$. The probability score for the observation vector sequence $\boldsymbol{X}$ and the model M_k can be calculated as the likelihood

$$\begin{aligned}lf_{X|\mathcal{M}}\big(\boldsymbol{X}\,|\mathcal{M}_k\big) &= \sum_{s}\pi_{s(0)}f_{X|S}\big(\boldsymbol{x}(0)\,|s(0)\big)a_{s(0)s(1)}f_{X|S}\big(\boldsymbol{x}(1)\,|s(1)\big)\\ &\quad\cdots a_{s(T-2)s(T-1)}f_{X|S}\big(x(T-1)\,|s(T-1)\big)\end{aligned} \tag{5.39}$$

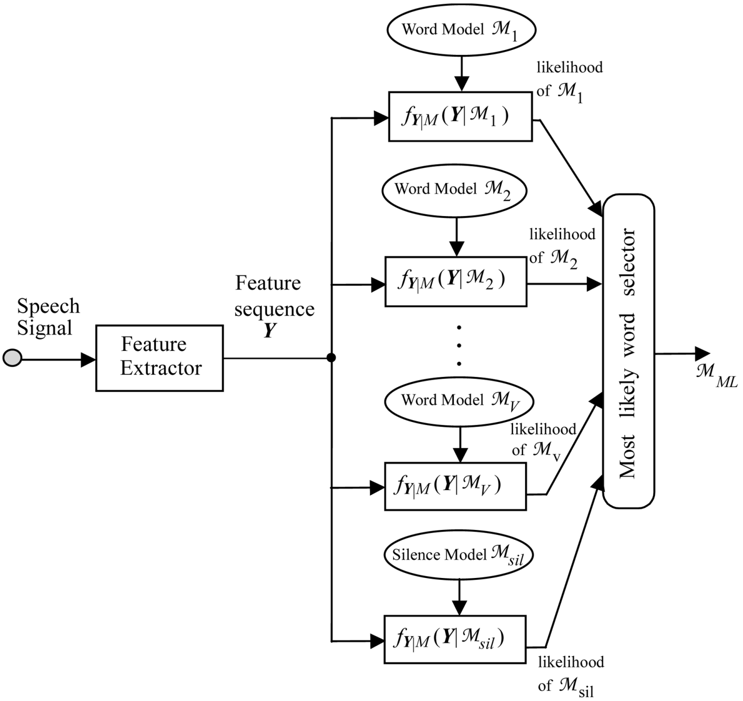

Speech Signal
Feature Extractor
Feature sequence Y
Word Model $\mathcal{M}_1$
$f_{Y|M}(Y|\mathcal{M}_1)$
likelihood of $\mathcal{M}_1$
Word Model $\mathcal{M}_2$
$f_{Y|M}(Y|\mathcal{M}_2)$
likelihood of $\mathcal{M}_2$
Word Model $\mathcal{M}_V$
$f_{Y|M}(Y|\mathcal{M}_V)$
likelihood of $\mathcal{M}_V$
Silence Model $\mathcal{M}_{sil}$
$f_{Y|M}(Y|\mathcal{M}_{sil})$
likelihood of $\mathcal{M}_{sil}$
Most likely word selector
$\mathcal{M}_{ML}$

Figure 5.10 Illustration of the use of HMMs in speech recognition.

where the likelihood of the observation sequence $\boldsymbol{X}$ is summed over all possible state sequences of the model M. Equation (5.39) can be efficiently calculated using the forward–backward method described in Section 5.3.1. The observation sequence $\boldsymbol{X}$ is labelled with the HMM that scores the highest likelihood as

$$\text{Label}(\boldsymbol{X}) = \arg\max_k \left(f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_k)\right), \quad k = 1, \ldots, V+1 \tag{5.40}$$

In decoding applications often the likelihood of an observation sequence $\boldsymbol{X}$ 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

$$\text{Label}(\boldsymbol{X}) = \arg\max_k \left[\max_s f_{X,S|\mathcal{M}}(\boldsymbol{X},\boldsymbol{s}|\mathcal{M}_k)\right] \quad k = 1, \ldots, V+1 \tag{5.41}$$

In Section 5.6, 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 maximum a posterior (MAP) estimate of the underlying state sequence. The MAP state sequence $\boldsymbol{s}^{\text{MAP}}$ of a model $\mathcal{M}$ given an observation signal sequence $\boldsymbol{X} = [\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]$ is obtained as

$$\begin{aligned} \boldsymbol{s}^{MAP} &= \arg\max_s f_{X,S|\mathcal{M}}(\boldsymbol{X}, \boldsymbol{s}|\mathcal{M}) \\ &= \arg\max_s \left(f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s}, \mathcal{M}) P_{S|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})\right) \end{aligned} \tag{5.42}$$

The MAP state sequence estimate is used in such applications as the calculation of a similarity score between a signal sequence $\boldsymbol{X}$ and an HMM $\mathcal{M}$, segmentation of a non-stationary signal into a number of distinct quasi-stationary segments, and implementation of state-based Wiener filters for restoration of noisy signals, as described in the Section 5.6.

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 next, and Figure 5.11 gives a network illustration of the algorithm.

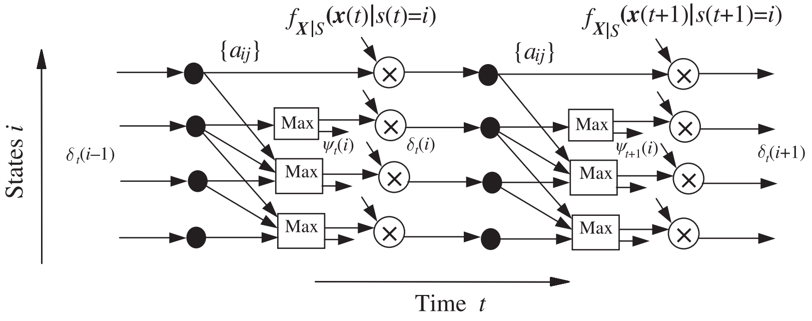


Figure 5.11 A network illustration of the Viterbi algorithm.

5.4.1.1 Viterbi Algorithm

$\delta_t(i)$ records the cumulative probability of the best path to state i at time t. $\psi_t(i)$ records the best state sequence to state i at time t.

Step 1: *Initialisation*, at time $t = 0$, for states $i = 1, \ldots, N$

$$\delta_0(i) = \pi_i f_i(\boldsymbol{x}(0))$$

$$\psi_0(i) = 0$$

Step 2: *Recursive calculation* of the ML state sequences and their probabilities
For time $t = 1, \ldots, T-1$
For states $i = 1, \ldots, N$

$$\delta_t(i) = \max_j \left[\delta_{t-1}(j)a_{ji}\right] f_i(\boldsymbol{x}(t))$$

$$\psi_t(i) = \arg\max_j \left[\delta_{t-1}(j)a_{ji}\right]$$

Step 3: *Termination*, retrieve the most likely final state

$$s^{\mathrm{MAP}}(T-1) = \arg\max_{i}[\delta_{T-1}(i)]$$

$$Prob_{\max} = \max_{i}\,[\delta_{T-1}(i)]$$

Step 4: *Backtracking* through the most likely state sequence:
For $t = T-2, \ldots, 0$

$$s^{MAP}(t) = \psi_{t+1}\left[s^{MAP}(t+1)\right].$$

The backtracking routine retrieves the most likely state sequence of the model $\mathcal{M}$. Note that the variable $Prob_{\max}$, which is the probability of the observation sequence $\boldsymbol{X} = [\boldsymbol{x}(0), \ldots, \ \boldsymbol{x}(T-1)]$ 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 $\boldsymbol{X}$. For example, in speech recognition, for each candidate word model the probability of the observation and the most likely state sequence is calculated, and then the observation is labelled with the word that achieves the highest probability score.

5.5 HMMs in DNA and Protein Sequences

A major application of hidden Markov models is in bio-signal 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 (HMMs) 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 modelling 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 sequence. Each column is a state for which the probabilities of occurrence of ACTG are calculated as the normalised 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 called *profile 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 proteins or DNA super families.

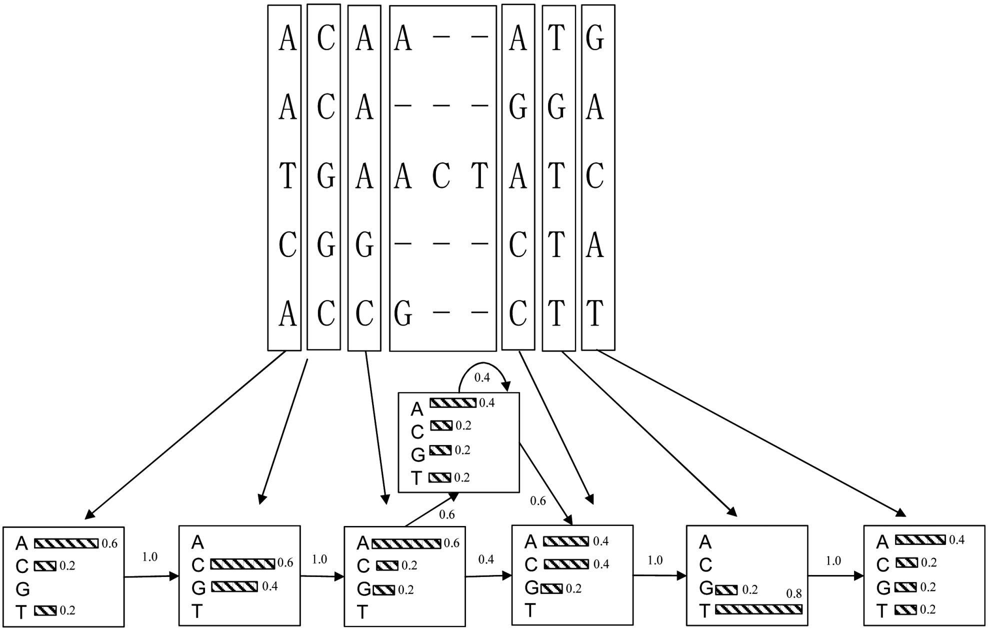


Figure 5.12 A Markov model for a dataset of DNA sequences. The discrete probabilities are histograms of the occurrence of each symbol in a column.

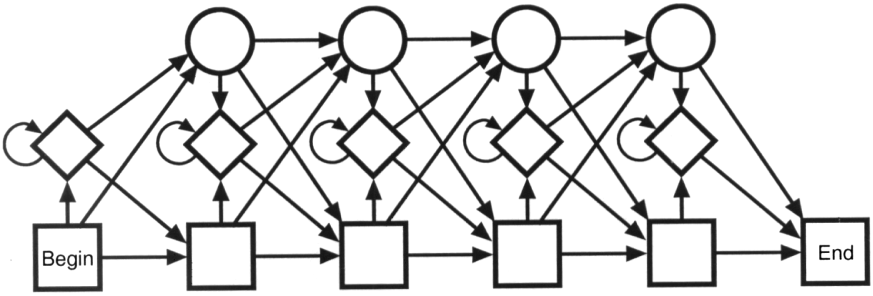


Figure 5.13 A DNA profile HMM: squares represent base states, diamonds are insertion states and circles are deletion states.

5.6 HMMs for Modelling Speech and Noise

5.6.1 *Modelling Speech*

HMMs are the main statistical modelling framework for speech recognition. Normally a three- to five-state 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. 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 and the following two sections, we consider the use of HMMs for estimation of a signal $\boldsymbol{x}(t)$ observed in an additive noise $\boldsymbol{n}(t)$, and modelled as

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + \boldsymbol{n}(t) \tag{5.43}$$

From Bayes' rule, the posterior pdf of the signal $\boldsymbol{x}(t)$ given the noisy observation $\boldsymbol{y}(t)$ is defined as

$$\begin{aligned} f_{X|Y}\big(\boldsymbol{x}(t)\,|\boldsymbol{y}(t)\,\big) &= \frac{f_{Y|X}\big(\boldsymbol{y}(t)\,|\boldsymbol{x}(t)\,\big)f_X(\boldsymbol{x}(t))}{f_Y(\boldsymbol{y}(t))} \\ &= \frac{1}{f_Y\big(\boldsymbol{y}(t)\big)} f_N\big(\boldsymbol{y}(t) - \boldsymbol{x}(t)\big) f_X\big(\boldsymbol{x}(t)\big) \end{aligned} \tag{5.44}$$

For a given observation, $f_Y(\boldsymbol{y}(t))$ is a constant, and the maximum a posteriori (MAP) estimate is obtained as

$$\hat{\boldsymbol{x}}^{\text{MAP}}(t) = \arg\max_{x(t)} \underbrace{f_N\big(\boldsymbol{y}(t) - \boldsymbol{x}(t)\big)}_{\text{Likelihood}} \underbrace{f_X\big(\boldsymbol{x}(t)\big)}_{\text{Prior}} \tag{5.45}$$

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-valued processes are often modelled by a Gaussian or a mixture Gaussian pdf that is equivalent to a single-state HMM. For a non-stationary process an N-state HMM can model the time-varying pdf of the process as a Markovian chain of N stationary Gaussian sub-processes. 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 the noise are uncorrelated, each signal state may be observed in any noisy state; therefore the number of noisy signal states is of the order of $N_s^T \times N_n^T$.

Given an observation sequence $\boldsymbol{Y} = [\boldsymbol{y}(0), \boldsymbol{y}(1), \ldots, \boldsymbol{y}(T-1)]$, the most probable state sequences of the signal and the noise HMMs maybe expressed as

$$\boldsymbol{s}_{\text{signal}}^{\text{MAP}} = \arg\max_{s_{\text{signal}}} \left(\max_{s_{\text{noise}}} f_Y\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} | \mathcal{M}, \eta\right)\right) \tag{5.46}$$

and

$$\boldsymbol{s}_{\text{noise}}^{MAP} = \arg\max_{s_{\text{noise}}} \left(\max_{s_{\text{signal}}} f_Y\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} | \mathcal{M}, \eta\right)\right) \tag{5.47}$$

Given the state sequence estimates for the signal and the noise models, the MAP estimation equation (5.45) becomes

$$\hat{\boldsymbol{x}}^{MAP}(t) = \arg\max_{\boldsymbol{x}} \left(f_{N|S,\eta}\left(\boldsymbol{y}(t) - \boldsymbol{x}(t)\,\left|\boldsymbol{s}_{\text{noise}}^{MAP}, \eta\right.\right) f_{X|S,\mathcal{M}}\left(\boldsymbol{x}(t)\,\left|\boldsymbol{s}_{\text{signal}}^{MAP}, \mathcal{M}\right.\right)\right) \tag{5.48}$$

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

$$\boldsymbol{y}(t) = \bar{s}(t)\boldsymbol{x}_0(t) + s(t)\boldsymbol{x}_1(t) + \boldsymbol{n}(t) \tag{5.49}$$

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 $\mathcal{N}(\boldsymbol{x}(t), \boldsymbol{\mu}_{x_0}, \boldsymbol{\Sigma}_{x_0x_0})$, and $s(t) = 1$ indicates that the signal is from the state S_1

with a Gaussian pdf of $\mathcal{N}(\boldsymbol{x}(t), \boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1x_1})$. Assume that a stationary Gaussian process $\mathcal{N}(\boldsymbol{n}(t), \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_{nn})$, equivalent to a single-state HMM, can model the noise. Using the Viterbi algorithm the maximum a posteriori (MAP) state sequence of the signal model can be estimated as

$$\boldsymbol{s}_{\text{signal}}^{\text{MAP}} = \arg\max_{s}\left[f_{Y|S,\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{s},\mathcal{M})P_{S|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})\right] \tag{5.50}$$

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

$$f_{Y|s_0}\big(\boldsymbol{y}(t)\,|s_0\big) = \mathcal{N}\big(\boldsymbol{y}(t), (\boldsymbol{\mu}_{\boldsymbol{x}_0} + \boldsymbol{\mu}_{\boldsymbol{n}}), (\boldsymbol{\Sigma}_{\boldsymbol{x}_0\boldsymbol{x}_0} + \boldsymbol{\Sigma}_{\boldsymbol{nn}})\big) \tag{5.51}$$

and

$$f_{Y|s_1}\big(\boldsymbol{y}(t)\,|s_1\big) = \mathcal{N}\big(\boldsymbol{y}(t), (\boldsymbol{\mu}_{\boldsymbol{x}_1} + \boldsymbol{\mu}_{\boldsymbol{n}}), (\boldsymbol{\Sigma}_{\boldsymbol{x}_1\boldsymbol{x}_1} + \boldsymbol{\Sigma}_{\boldsymbol{nn}})\big) \tag{5.52}$$

where $\mathcal{N}(\boldsymbol{y}(t), \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The MAP signal estimate, given a state sequence estimate $\boldsymbol{s}^{\text{MAP}}$, is obtained from

$$\hat{\boldsymbol{x}}^{\text{MAP}}(t) = \arg\max_{\boldsymbol{x}}\left[f_{X|S,\mathcal{M}}\big(\boldsymbol{x}(t)|\boldsymbol{s}^{\text{MAP}},\mathcal{M}\big)f_N\big(\boldsymbol{y}(t) - \boldsymbol{x}(t)\big)\right] \tag{5.53}$$

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:

$$\hat{\boldsymbol{x}}^{MAP}(t) = \big(\boldsymbol{\Sigma}_{xx,s(t)} + \boldsymbol{\Sigma}_{nn}\big)^{-1}\boldsymbol{\Sigma}_{xx,s(t)}\big(\boldsymbol{y}(t) - \boldsymbol{\mu}_n\big) + \big(\boldsymbol{\Sigma}_{xx,s(t)} + \boldsymbol{\Sigma}_{nn}\big)^{-1}\boldsymbol{\Sigma}_{nn}\boldsymbol{\mu}_{x,s(t)} \tag{5.54}$$

where $\boldsymbol{\mu}_{x,s(t)}$ and $\boldsymbol{\Sigma}_{xx,s(t)}$ are the mean vector and covariance matrix of the signal $\boldsymbol{x}(t)$ 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

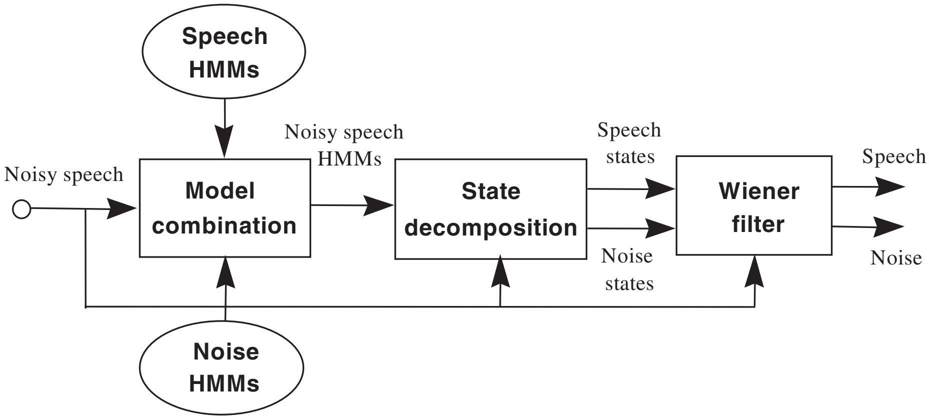


Figure 5.14 Outline configuration of HMM-based noisy speech recognition and enhancement.

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.

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 12 and 15) or by combining the noise and the signal models to model the noisy signal.

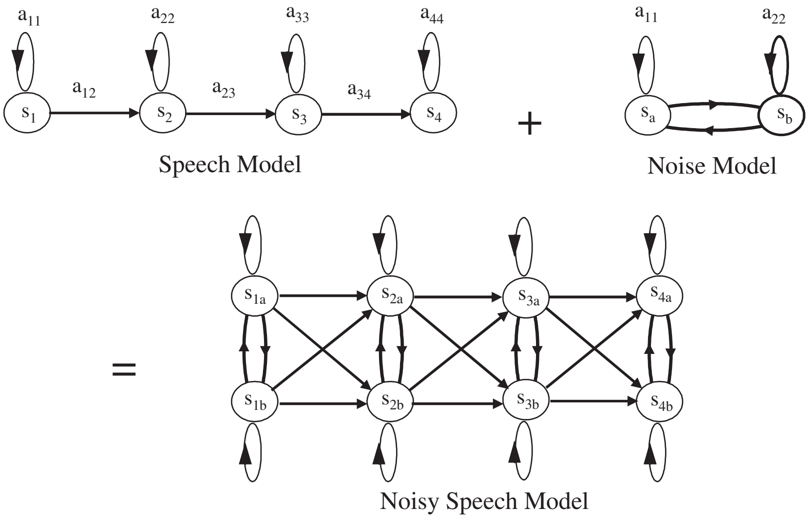


Figure 5.15 Outline configuration of HMM-based noisy speech recognition and enhancement. S_{ij} is a combination of the state i of speech with the state j of noise.

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 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:

$$\boldsymbol{\mu}_y = \boldsymbol{\mu}_x + g\boldsymbol{\mu}_n \tag{5.55}$$

$$\boldsymbol{\Sigma}_{yy} = \boldsymbol{\Sigma}_{xx} + g^2 \boldsymbol{\Sigma}_{nn} \tag{5.56}$$

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.

Note that in the combined model of Figure 5.15 as the number of states and the number of transitions from each state increases, compared with the individual constituent models, the state transition probability will also be affected. For example, the probability of a transition from state $\mathrm{s}_{1\mathrm{a}}$ to $\mathrm{s}_{2\mathrm{b}}$ is the probability that the speech will move from state 1 to state 2 and the probability that independently the noise will move from state *a* to state *b*.

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 the noisy speech from the ML combined model.
(3) Extract the signal and noise states from the ML state sequence of the ML combined noisy signal model obtained in (2).

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

$$\boldsymbol{w} = [\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn}]^{-1}\boldsymbol{r}_{xx} \tag{5.57}$$

and

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{5.58}$$

where $\boldsymbol{R}_{xx}$, $\boldsymbol{r}_{xx}$ 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.

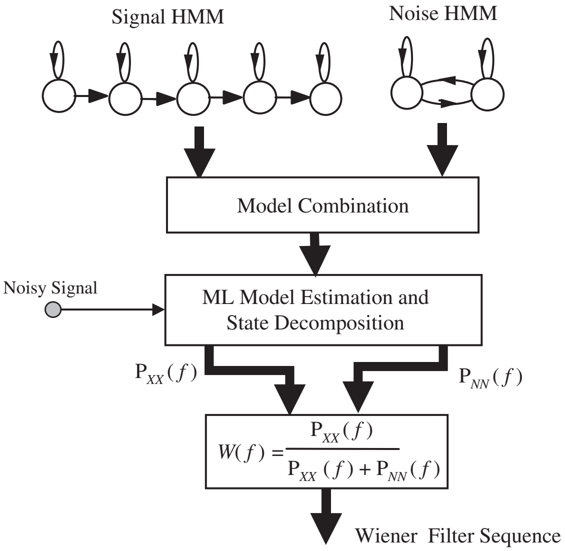


Figure 5.16 Illustration of HMMs with state-dependent Wiener filters.

Assuming that there are V HMMs $\{\mathcal{M}_1, \ldots, \mathcal{M}_V\}$ 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.

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 non-stationary noise, a multi-state HMM can model the time variations of the noise process with a finite number of quasi-stationary states. In general, the number of states required to accurately model the noise depends on the non-stationary character of the noise.

An example of a non-stationary 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

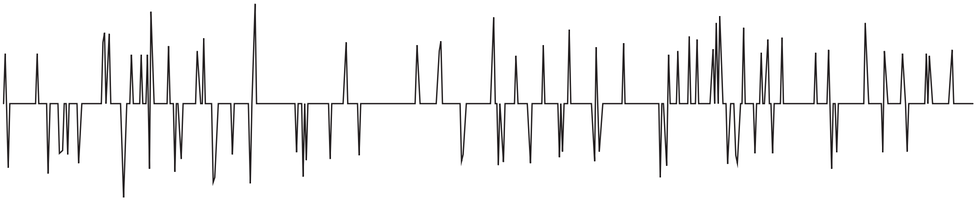


Figure 5.17 Impulsive noise.

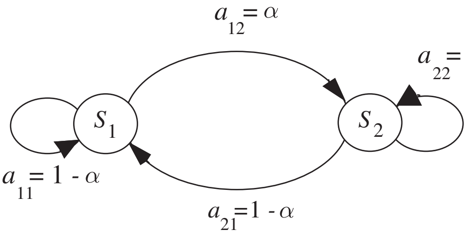


Figure 5.18 A binary-state model of an impulsive noise process.

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 multi-state HMM to model the pulse itself. HMMs are used for modelling impulsive noise and for channel equalisation, see Chapter 16.

5.7 Summary

HMMs provide a powerful method for the modelling of non-stationary 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 non-stationary process with a chain of stationary sub-processes. 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 training 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 decoding 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, decoding the underlying state sequence and labelling the observation with one of a number of candidate HMMs are accomplished using the Viterbi method. In Section 5.6 we considered the use of HMMs for MAP estimation of a signal observed in noise, and considered the use of HMMs in implementation of state-based Wiener filter sequences.

Bibliography

Bahl L.R., Jelinek F. and Mercer R.L. (1983) A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **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 Proc. Acoustics, Speech and Signal Processing*, **ICASSP-86**, Tokyo: 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. Stat.*, **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. Stat.*, **41**: 164–171.

Conner P.N. (1993) Hidden Markov Model with Improved Observation and Duration Modelling. PhD Thesis, University of East Anglia, England.

Epharaim Y., Malah D. and Juang B.H.(1989) On Application of Hidden Markov Models for Enhancing Noisy Speech. *IEEE Trans. Acoustics Speech and Signal Processing*, **37**(12): 1846–1856, Dec.

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. *Proc. IEEE, Int. Conf. on Acoust., Speech, Signal Processing*, **ICASSP-92**: 233–236.

Gales M.J.F. and Young S.J. (1993) HMM Recognition in Noise using Parallel Model Combination. *Eurospeech-93*: 837–840.

Huang X.D. and Jack M.A. (1989) Unified Techniques for Vector Quantisation and Hidden Markov Modelling using Semi-Continuous Models. *IEEE Proc. Acoustics, Speech and Signal Processing*, **ICASSP-89**, Glasgow: 639–642.

Huang X.D., Ariki Y. and Jack M.A. (1990) *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, Edinburgh.

Jelinek F. and Mercer R. (1980) Interpolated Estimation of Markov Source Parameters from Sparse Data. *Proc. of the Workshop on Pattern Recognition in Practice*. North-Holland, Amsterdam.

Jelinek F. (1976) Continuous Speech Recognition by Statistical Methods. *Proc. of IEEE*, **64**: 532–556.

Juang B.H. (1984) On the Hidden Markov Model and Dynamic Time Warping for Speech Recognition – A unified Overview. *AT&T Technical J*., **63**: 1213–1243.

Juang B.H. (1985) Maximum-Likelihood Estimation for Mixture Multi-Variate Stochastic Observations of Markov Chain. *AT&T Bell Laboratories Tech J*., **64**: 1235–1249.

Kullback S. and Leibler R.A. (1951) On Information and Sufficiency. *Ann. Math. Stat*., **22**: 79–86.

Lee K.F. (1989) *Automatic Speech Recognition: the Development of SPHINX System*. Kluwer Academic Publishers, Boston, MA.

Lee K.F. (1989) 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-82**: 729–734.

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, PhD Thesis, University of East Anglia, England.

Peterie T. (1969) Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Stat*., **40**: 97–115.

Rabiner L.R. and Juang B.H. (1986) An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*: 4–16.

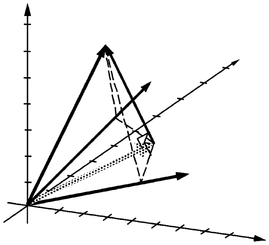
Rabiner L.R. and Juang B.H. (1993) *Fundamentals of Speech Recognition*. Prentice-Hall, Englewood Cliffs, NJ.

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 Tech. J*., **64**: 1211–1234.

Varga A. and Moore R.K. (1990) Hidden Markov Model Decomposition of Speech and Noise. *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Processing*: 845–848.

Viterbi A.J. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Information Theory*, **IT-13**: 260–226.

Young S.J. (1999) *HTK: Hidden Markov Model Tool Kit*. Cambridge University Engineering Department.



6

Least Square Error Wiener-Kolmogorov Filters

Least squared error filter theory was initially formulated independently by Andrei Kolmogorov (1941) and Norbert Wiener (1949). While Kolmogorov's method was based on time-domain analysis, Wiener's method was based on frequency domain analysis.

Least squared error filter theory forms the foundation of data-dependent adaptive linear filters. Least squared error filters play a central role in a wide range of applications such as linear prediction, echo cancellation, signal restoration, channel equalisation, radar signal processing and system identification.

The coefficients of a least squared error filter are calculated to minimise the average squared distance between the filter output and a desired or target signal. The solution for the Wiener filter coefficients requires estimates of the autocorrelation of the input and the cross-correlation of the input and the desired signal.

In its basic form, the least squared 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 each block and becomes block-adaptive. A block-adaptive (or frame-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 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-Kolmogorov Filter

Norbert Wiener, and independently Andrei Nikolaevich 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.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

The typical scenarios in which Wiener filters are used are in the contexts of estimation or prediction of a signal observed in noise and system identification/estimation (such as channel estimation) given the inputs and the outputs of a system.

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 telecommunication channel, or noise reduction. The Wiener filter can also be used to predict the trajectory of a projectile; a problem during the second world war on which Norbert 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 squared 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.

6.1.1 Derivation of Wiener Filter Equation

Figure 6.1 illustrates a Wiener filter represented by the filter's coefficient vector $\boldsymbol{w}$. The filter takes as the input a signal $y(m)$, usually a distorted version of a desired signal $x(m)$, 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

$$\begin{aligned}\hat{x}(m) &= \sum_{k=0}^{P-1} w_k y(m-k)\\ &= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}\end{aligned} \tag{6.1}$$

where m is the discrete-time index, vector $\boldsymbol{y}^{\mathrm{T}} = [y(m),\ y(m-1), \ldots,\ y(m-P-1)]$ is the filter input signal, $\hat{x}(m)$ is the filter output and the parameter vector $\boldsymbol{w}^{\mathrm{T}} = [w_0,\ w_1, \ldots,\ w_{P-1}]$ 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.

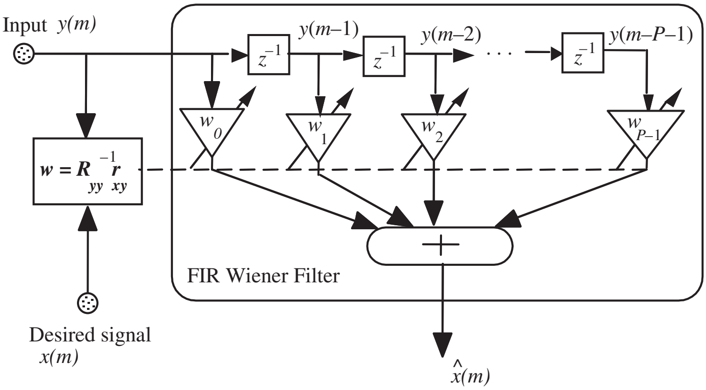


Figure 6.1 Illustration of a Wiener filter. The output signal $\hat{x}(m)$ is an estimate of the desired signal $x(m)$. It is obtained as the product of the input vector $[y(m-1)\ldots y(m-P-1)]$ and the coefficients vector $[w_o \ldots w_{P-1}]$.

The Wiener filter error signal, $e(m)$ is defined as the difference between the desired (or target) signal $x(m)$ and the filter output $\hat{x}(m)$:

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{y} \end{aligned} \tag{6.2}$$

where, as expressed in Equation (6.1), $\hat{x}(m)$ is the convolution of the input signal vector $\boldsymbol{y}$ and Wiener filter $\boldsymbol{w}$. 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 $\boldsymbol{w}$. 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 $\boldsymbol{w}$ and the error signal $e(m)$ we write Equation (6.2) N times in a matrix for a segment of N samples of the target (or desired) signal $[x(0), x(1), \ldots, x(N-1)]$ and the observation (or input) signal $[y(0), y(1), \ldots, y(N-1)]$ as,

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} y(0) & y(-1) & y(-2) & \cdots & y(1-P) \\ y(1) & y(0) & y(-1) & \cdots & y(2-P) \\ y(2) & y(1) & y(0) & \cdots & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & y(N-3) & \cdots & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} \tag{6.3}$$

In a compact vector notation this matrix equation may be written as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w} \tag{6.4}$$

where $\boldsymbol{e}$ is the error vector, $\boldsymbol{x}$ is the desired signal vector, $\boldsymbol{Y}$ is the input signal matrix and $\boldsymbol{Y}\boldsymbol{w} = \hat{\boldsymbol{x}}$ is the Wiener filter output signal vector. It is assumed that the P initial input signal samples $[y(-1), \ldots, y(-P-1)]$ 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 (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 $\boldsymbol{w}$, with a zero estimate on error $\boldsymbol{e} = \boldsymbol{0}$, such that $\hat{\boldsymbol{x}} = \boldsymbol{Y}\boldsymbol{w} = \boldsymbol{x}$.

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 *under-determined*.

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 *over-determined* and has a unique solution, usually with a non-zero 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 $\mathcal{E}\,[.]$ 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 a linear and closed-form solution. 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 $\boldsymbol{w}$, where $\mathcal{E}$ is expectation or average. From Equation (6.2), the mean square estimation error is given by

$$\begin{aligned}\mathcal{E}\left[e^2(m)\right] &= \mathcal{E}\left[\left(x(m)-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}\right)^2\right]\\ &= \mathcal{E}\left[x^2(m)\right] - 2\boldsymbol{w}^{\mathrm{T}}\mathcal{E}[\boldsymbol{y}x(m)] + \boldsymbol{w}^{\mathrm{T}}\mathcal{E}\left[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right]\boldsymbol{w}\\ &= r_{xx}(0) - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{r}_{yx} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\,\boldsymbol{w}\end{aligned} \tag{6.5}$$

where $\boldsymbol{R}_{yy} = E\,[\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)]$ is the autocorrelation matrix of the input signal and $\boldsymbol{r}_{xy} = E\,[x(m)\boldsymbol{y}(m)]$ is the cross-correlation vector of the input and the desired signals. An expanded form of Equation (6.5) can be obtained as

$$\mathcal{E}[e^2(m)] = r_{xx}(0) - 2\sum_{k=0}^{P-1} w_k r_{yx}(k) + \sum_{k=0}^{P-1} w_k \sum_{j=0}^{P-1} w_j\, r_{yy}(k-j) \tag{6.6}$$

where $r_{yy}(k)$ and $r_{yx}(k)$ are the elements of the autocorrelation matrix $\boldsymbol{R}_{yy}$ and the cross-correlation vector $\boldsymbol{r}_{xy}$ respectively.

From Equation (6.5), the mean square error for an FIR filter is a quadratic function of the filter coefficient vector $\boldsymbol{w}$ 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

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{w}}\ \mathcal{E}[e^2(m)] &= -2\mathcal{E}[x(m)\boldsymbol{y}(m)] + 2\boldsymbol{w}^{\mathrm{T}}\mathcal{E}[\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)]\\ &= -2\boldsymbol{r}_{yx} + 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\end{aligned} \tag{6.7}$$

where the gradient vector is defined as

$$\frac{\partial}{\partial \boldsymbol{w}} = \left[\frac{\partial}{\partial w_0},\ \frac{\partial}{\partial w_1},\ \frac{\partial}{\partial w_2},\ldots,\ \frac{\partial}{\partial w_{P-1}}\right]^{\mathrm{T}} \tag{6.8}$$

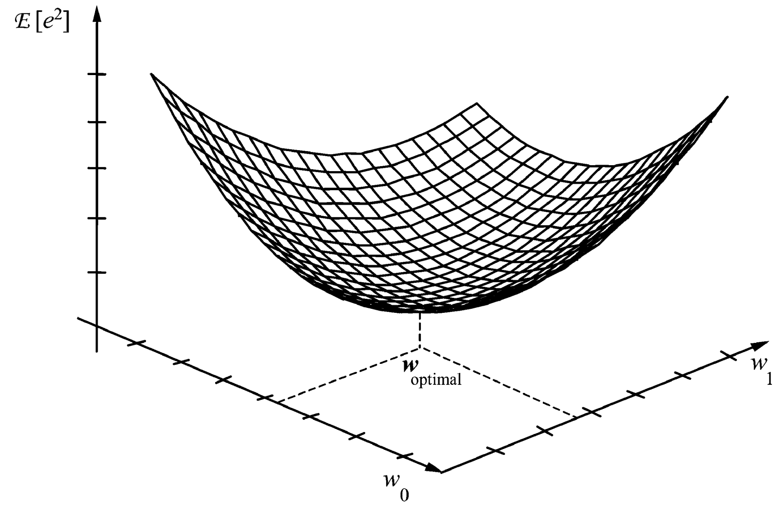


Figure 6.2 Mean square error surface for a two-tap FIR filter.

The minimum mean square error Wiener filter is obtained by setting Equation (6.7) to zero:

$$\boldsymbol{R}_{yy}\,\boldsymbol{w}=\boldsymbol{r}_{yx} \tag{6.9}$$

or, equivalently,

$$\boldsymbol{w}=\boldsymbol{R}_{yy}^{-1}\,\boldsymbol{r}_{yx} \tag{6.10}$$

In an expanded form, the Wiener filter solution Equation (6.10) can be written as

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} = \begin{pmatrix} r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \cdots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \cdots & r_{yy}(P-2) \\ r_{yy}(2) & r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & r_{yy}(P-3) & \cdots & r_{yy}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{yx}(0) \\ r_{yx}(1) \\ r_{yx}(2) \\ \vdots \\ r_{yx}(P-1) \end{pmatrix} \tag{6.11}$$

6.1.2 Calculation of Autocorrelation of Input and Cross-Correlation of Input and Desired Signals

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 non-stationary and need to be segmented in quasi-stationary 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

$$r_{yy}(k)=\frac{1}{N}\sum_{m=0}^{N-1}y(m)y(m+k) \tag{6.12}$$

Note from Equation (6.11) that the autocorrelation matrix $\boldsymbol{R}_{yy}$ 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.

The cross-correlation values are estimated as

$$r_{yx}(k)=\frac{1}{N}\sum_{m=0}^{N-1}y(m)x(m+k) \tag{6.13}$$

The calculation of the autocorrelation of the input signal $r_{yy}(k)$ using Equation (6.12) is straightforward. The main challenge is in the calculation of the cross-correlation $r_{yx}(k)$ since the signal $x(m)$ is not available. However, this problem is resolved by either using pre-calculated values or taking advantage of the specific form of the problem as in Section 6.6.

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 $[y(0),\ y(1),\ \ldots,\ y(N-1)]$ and the desired signal $[x(0),\ x(1),\ \ldots,\ x(N-1)]$. The set of N linear equations describing the input/output relationship of Wiener filter can be written in matrix form as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = \begin{pmatrix} y(0) & y(-1) & y(-2) & \cdots & y(2-P) & y(1-P) \\ y(1) & y(0) & y(-1) & \cdots & y(3-P) & y(2-P) \\ y(2) & y(1) & y(0) & \cdots & y(4-P) & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(N-2) & y(N-3) & y(N-4) & \cdots & y(N-P) & y(N-1-P) \\ y(N-1) & y(N-2) & y(N-3) & \cdots & y(N+1-P) & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-2} \\ w_{P-1} \end{pmatrix} \tag{6.14}$$

Equation (6.14) can be rewritten in compact matrix notation as

$$\hat{\boldsymbol{x}} = \boldsymbol{Y}\boldsymbol{w} \tag{6.15}$$

The Wiener filter error is the difference between the desired signal and the filter output defined as

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{x} - \hat{\boldsymbol{x}} \\ &= \boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w} \end{aligned} \tag{6.16}$$

The energy of the error vector, that is the sum of the squared elements of the error vector $\boldsymbol{e}$, is given by the inner vector product as

$$\begin{aligned} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} &= (\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w}) \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{w} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{w} \end{aligned} \tag{6.17}$$

The gradient of the squared error function with respect to the Wiener filter coefficients is obtained by differentiating Equation (6.17) w.r.t $\boldsymbol{w}$ as

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{w}} = -2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Y} + 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} \tag{6.18}$$

The Wiener filter coefficients are obtained by setting the gradient of the squared error function of Equation (6.18) to zero; this yields

$$\left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)\boldsymbol{w} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} \tag{6.19}$$

or

$$\boldsymbol{w} = \left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} \tag{6.20}$$

Note that the matrix $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$ is a time-averaged estimate of the autocorrelation matrix of the filter input signal $\boldsymbol{R}_{yy}$, and that the vector $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x}$ is a time-averaged estimate of $\boldsymbol{r}_{xy}$ 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 of 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 (6.20) approaches the Wiener filter of Equation (6.10):

$$\lim_{N\to\infty}\left[\boldsymbol{w} = \left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x}\right] = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{xy} \tag{6.21}$$

6.2.1 QR Decomposition of the Least Square Error Equation

An efficient and robust method for solving the least square error Equation (6.20) is the QR decomposition (QRD) method. In this method, the $N \times P$ signal matrix $\boldsymbol{Y}$ (shown in Equation (6.14)) is decomposed into the product of an $N \times N$ orthonormal matrix $\boldsymbol{Q}$ and a $P \times P$ upper-triangular matrix $\boldsymbol{R}$ as

$$\boldsymbol{QY} = \begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix} \tag{6.22}$$

where $\mathbf{0}$ is the $(N-P) \times P$ null matrix, $\boldsymbol{Q}$ is an orthonormal matrix

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}} = \mathbf{I} \tag{6.23}$$

and the upper-triangular matrix $\boldsymbol{R}$ is of the form

$$\boldsymbol{R} = \begin{pmatrix} r_{00} & r_{01} & r_{02} & r_{03} & \cdots & r_{0P-1} \\ 0 & r_{11} & r_{12} & r_{13} & \cdots & r_{1P-1} \\ 0 & 0 & r_{22} & r_{23} & \cdots & r_{2P-1} \\ 0 & 0 & 0 & r_{33} & \cdots & r_{3P-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & r_{P-1P-1} \end{pmatrix} \tag{6.24}$$

From Equations (6.22) and (6.23) we have

$$\boldsymbol{Y} = \boldsymbol{Q}^{T} \begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix} \tag{6.25}$$

Substitution of Equation (6.25) in Equation (6.19) yields

$$\begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix}^{\mathrm{T}} \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix} \boldsymbol{w} = \begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix}^{\mathrm{T}} \boldsymbol{Q}\,\boldsymbol{x} \tag{6.26}$$

From Equation (6.26) we have

$$\begin{pmatrix} \boldsymbol{R} \\ \mathbf{0} \end{pmatrix} \boldsymbol{w} = \boldsymbol{Qx} \tag{6.27}$$

From Equation (6.27) we have

$$\boldsymbol{Rw} = \boldsymbol{x}_Q \tag{6.28}$$

where the vector $\boldsymbol{x}_Q$ on the right-hand side of Equation (6.28) is composed of the first P elements of the product $\boldsymbol{Qx}$. Since the matrix $\boldsymbol{R}$ is upper-triangular, the coefficients of the least square error filter can be obtained easily through a process of back substitution from Equation (6.28), 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 $\boldsymbol{Q}$ and of the upper triangular matrix $\boldsymbol{R}$. The decomposition of a matrix into $\boldsymbol{QR}$ 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 Filter as Projection in Vector Space

In this section, we consider an alternative formulation of Wiener filters where the least square error estimate is visualised 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 $\hat{\boldsymbol{x}}$ as a linear weighted combination of the column vectors of the input signal matrix as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = w_0 \begin{pmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N-2) \\ y(N-1) \end{pmatrix} + w_1 \begin{pmatrix} y(-1) \\ y(0) \\ y(1) \\ \vdots \\ y(N-3) \\ y(N-2) \end{pmatrix} + \cdots + w_{P-1} \begin{pmatrix} y(1-P) \\ y(2-P) \\ y(3-P) \\ \vdots \\ y(N-1-P) \\ y(N-P) \end{pmatrix} \tag{6.29}$$

In compact notation, Equation (6.29) may be written as

$$\hat{\boldsymbol{x}} = w_0 \boldsymbol{y}_0 + w_1 \boldsymbol{y}_1 + \cdots + w_{P-1} \boldsymbol{y}_{P-1} \tag{6.30}$$

In Equation (6.30) the Wiener filter output $\hat{\boldsymbol{x}}$ is expressed as a linear combination of P basis vectors $[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{P-1}]$, and hence it can be said that the estimate $\hat{\boldsymbol{x}}$ is in the vector subspace formed by the input signal vectors $[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{P-1}]$.

In general, the N-dimensional input signal vectors $[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{P-1}]$ in Equation (6.30) 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 $\boldsymbol{x}$. In this case, the signal estimate $\hat{\boldsymbol{x}} = \boldsymbol{x}$ and the estimation error is zero. However, in practice, $N > P$, and the signal space defined by the P input signal vectors of Equation (6.30) is only a subspace of the N-dimensional signal space. In this case, the estimation error is zero only if the desired signal $\boldsymbol{x}$ happens to be in the subspace of the input signal, otherwise the best estimate of $\boldsymbol{x}$ is the perpendicular projection of the vector $\boldsymbol{x}$ onto the vector space of the input signal $[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{P-1}]$, 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 $\boldsymbol{y}^{\mathrm{T}} = [y(m), y(m-1), y(m-2), y(m-3)]$ is the input observation signal, $\boldsymbol{x}^{\mathrm{T}} = [x(m), x(m-1), x(m-2)]$ is the desired signal and $\boldsymbol{w}^{\mathrm{T}} = [w_0, w_1]$ is the filter coefficient vector. As in Equation (6.27), the filter output can be written as

$$\begin{pmatrix} \hat{x}(m) \\ \hat{x}(m-1) \\ \hat{x}(m-2) \end{pmatrix} = w_0 \begin{pmatrix} y(m) \\ y(m-1) \\ y(m-2) \end{pmatrix} + w_1 \begin{pmatrix} y(m-1) \\ y(m-2) \\ y(m-3) \end{pmatrix} \tag{6.31}$$

In Equation (6.29), the filter input signal vectors $\boldsymbol{y}_1^{\mathrm{T}} = [y(m),\ y(m-1),\ y(m-2)]$ and $\boldsymbol{y}_2^{\mathrm{T}} = [y(m-1),\ y(m-2),\ y(m-3)]$ are 3-dimensional vectors. The subspace defined by the linear combinations of the two input vectors $[\boldsymbol{y}_1, \boldsymbol{y}_2]$ is a 2-dimensional plane in a 3-dimensional space. The filter output is a linear combination of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, and hence it is confined to the plane containing these two vectors. The least square error estimate of $\boldsymbol{x}$ is the orthogonal projection of $\boldsymbol{x}$ on the plane of $[\boldsymbol{y}_1, \boldsymbol{y}_2]$ as shown by the shaded vector $\hat{\boldsymbol{x}}$. If the desired vector happens to be in the plane defined by the vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ then the estimation error will be zero, otherwise the estimation error will be the perpendicular distance of $\boldsymbol{x}$ from the plane containing $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.

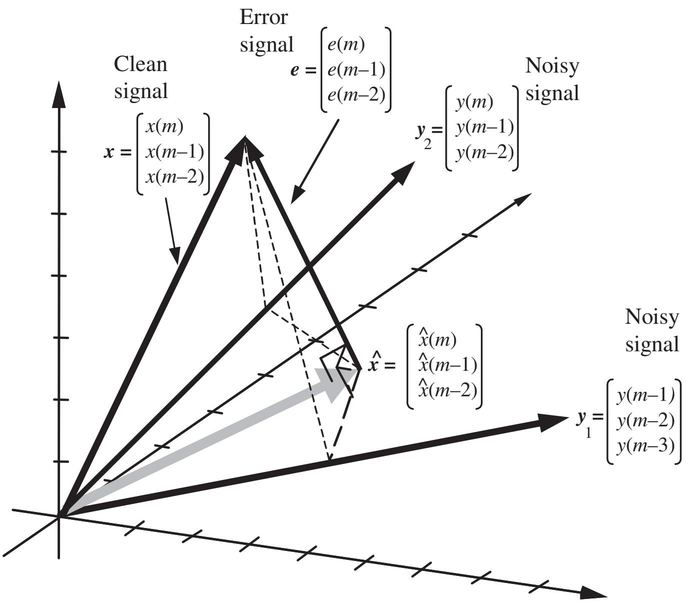

Error signal
$e = \begin{bmatrix} e(m) \\ e(m-1) \\ e(m-2) \end{bmatrix}$
Clean signal
$x = \begin{bmatrix} x(m) \\ x(m-1) \\ x(m-2) \end{bmatrix}$
Noisy signal
$y_2 = \begin{bmatrix} y(m) \\ y(m-1) \\ y(m-2) \end{bmatrix}$
$\hat{x} = \begin{bmatrix} \hat{x}(m) \\ \hat{x}(m-1) \\ \hat{x}(m-2) \end{bmatrix}$
Noisy signal
$y_1 = \begin{bmatrix} y(m-1) \\ y(m-2) \\ y(m-3) \end{bmatrix}$

Figure 6.3 The least square error projection of a desired signal vector $\boldsymbol{x}$ onto a plane containing the input signal vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is the perpendicular projection of $\boldsymbol{x}$ 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 $\boldsymbol{R}_{yy}\boldsymbol{w}=\boldsymbol{r}_{yx}$ in the mean squared error Equation (6.5) gives the least mean square error

$$\begin{aligned}\mathcal{E}[e^2(m)] &= r_{xx}(0) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{r}_{yx} \\ &= r_{xx}(0) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w}\end{aligned} \tag{6.32}$$

Now, for zero-mean signals, it is easy to show that the term $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w}$ in Equation (6.32) is the variance of the Wiener filter output $\hat{x}(m)$:

$$\sigma_{\hat{x}}^2 = \mathcal{E}[\hat{x}^2(m)] = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w} \tag{6.33}$$

Therefore Equation (6.32) may be written as

$$\sigma_e^2 = \sigma_x^2 - \sigma_{\hat{x}}^2 \tag{6.34}$$

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 $n(m)$:

$$y(m) = x_c(m) + n(m) \tag{6.35}$$

where the signal $x_c(m)$ is that part of the observation $y(m)$ that is correlated with the desired signal $x(m)$; it is this part of the input signal that may be transformable through a Wiener filter to the desired signal. Using Equation (6.35) the Wiener filter error may be decomposed into two distinct components:

$$\begin{aligned} e(m) &= x(m) - \sum_{k=0}^{P} w_k y(m-k) \\ &= \left[x(m) - \sum_{k=0}^{P} w_k x_c(m-k) \right] - \sum_{k=0}^{P} w_k n(m-k) \end{aligned} \tag{6.36}$$

or

$$e(m) = e_x(m) + e_n(m) \tag{6.37}$$

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.

$$e_x(m) = x(m) - \sum_{k=0}^{P-1} w_k x_c(m-k) \tag{6.38}$$

and $e_n(m)$ is the error in the filter output due to the presence of noise $n(m)$ in the input signal:

$$e_n(m) = -\sum_{k=0}^{P-1} w_k \, n(m-k) \tag{6.39}$$

The variance of filter error can be rewritten as

$$\sigma_e^2 = \sigma_{e_x}^2 + \sigma_{e_n}^2 \tag{6.40}$$

Note that in Equation (6.37), $e_x(m)$ 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. The issue of signal and noise separability is addressed in Section 6.6.2.
(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.

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 $W(f)$:

$$\hat{X}(f) = W(f)Y(f) \tag{6.41}$$

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

$$\begin{aligned} E(f) &= X(f) - \hat{X}(f) \\ &= X(f) - W(f)Y(f) \end{aligned} \tag{6.42}$$

and the mean square error at a frequency f is given by

$$\mathcal{E}\left[|E(f)|^2\right] = \mathcal{E}[(X(f) - W(f)Y(f))^* \, (X(f) - W(f)Y(f))] \tag{6.43}$$

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

$$\sum_{m=0}^{N-1} e^2(m) = \int_{-1/2}^{1/2} |E(f)|^2 \, df \tag{6.44}$$

To obtain the least mean square error filter we set the complex derivative of Equation (6.43) with respect to filter $W(f)$ to zero

$$\frac{\partial \, \mathcal{E}[|E(f)|^2]}{\partial W(f)} = 2W(f)P_{YY}(f) - 2P_{XY}(f) = 0 \tag{6.45}$$

where $P_{YY}(f) = E[Y(f)Y^*(f)]$ and $P_{XY}(f) = E[X(f)Y^*(f)]$ are the power spectrum of $Y(f)$, and the cross-power spectrum of $Y(f)$ and $X(f)$ respectively. From Equation (6.45), the least mean square error Wiener filter in the frequency domain is given as

$$W(f) = \frac{P_{XY}(f)}{P_{YY}(f)} \tag{6.46}$$

Alternatively, the frequency Wiener filter Equation (6.46) can be obtained from the Fourier transform of the time-domain Wiener Equation (6.9):

$$\sum_m \sum_{k=0}^{P-1} w_k r_{yy}(m-k)e^{-j\omega m} = \sum_m r_{yx}(m)e^{-j\omega m} \tag{6.47}$$

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 (6.47) can be transformed into frequency as

$$W(f)\; P_{YY}(f) = P_{XY}(f) \tag{6.48}$$

Re-arrangement of Equation (6.48) gives Equation (6.46).

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 multi-channel or multi-sensor systems, and in communication channel equalisation.

6.6.1 Wiener Filter for Additive Noise Reduction

Consider a signal $x(m)$ observed in a broadband additive noise $n(m)$, and modelled as

$$y(m) = x(m) + n(m) \tag{6.49}$$

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 $n(m)$:

$$\boldsymbol{R_{yy}} = \boldsymbol{R_{xx}} + \boldsymbol{R_{nn}} \tag{6.50}$$

and we can also write

$$\boldsymbol{r}_{xy} = \boldsymbol{r}_{xx} \tag{6.51}$$

where $\boldsymbol{R}_{yy}$, $\boldsymbol{R}_{xx}$ and $\boldsymbol{R}_{nn}$ are the autocorrelation matrices of the noisy signal, the noise-free signal and the noise respectively, and $\boldsymbol{r}_{xy}$ is the cross-correlation vector of the noisy signal and the noise-free signal. Substitution of Equations (6.50) and (6.51) in the Wiener filter, Equation (6.10), yields

$$\boldsymbol{w} = (\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})^{-1}\boldsymbol{r}_{xx} \tag{6.52}$$

Equation (6.52) 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

$$Y(f) = X(f) + N(f) \tag{6.53}$$

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

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{6.54}$$

where $P_{XX}(f)$ and $P_{NN}(f)$ are the signal and noise power spectra. Dividing the numerator and the denominator of Equation (6.54) by the noise power spectra $P_{NN}(f)$ and substituting the variable $SNR(f) = P_{XX}(f)/P_{NN}(f)$ yields

$$W(f) = \frac{SNR(f)}{SNR(f) + 1} \tag{6.55}$$

where SNR is a signal-to-noise ratio measure. Note that in Equation (6.55) the variable $SNR(f)$ is expressed in terms of the power-spectral ratio, and not in the more usual terms of log power ratio or dB. Therefore $SNR(f) = 0$ corresponds to zero signal content or $10\log_{10}(0) = -\infty$ dB and $SNR(f) = 1$ corresponds to equal signal and noise power $P_{XX}(f) = P_{NN}(f)$ or $10\log_{10}(1) = 0\,\text{dB}$ and $SNR(f) = 0.5$ corresponds to $10\log_{10}(0.5) = -3\,\text{dB}$. Figure 6.4 shows the variation of the Wiener filter response $W(f)$, with the signal-to-noise ratio $SNR(f)$.

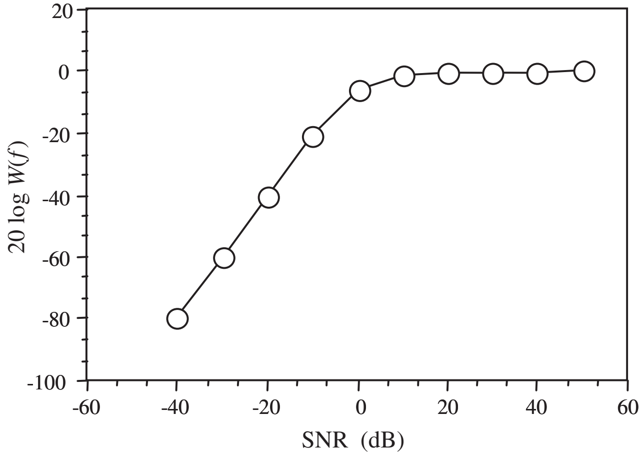


Figure 6.4 Variation of the gain of Wiener filter frequency response with SNR.

From Equation (6.55), 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 $0 \le W(f) \le 1$. Now consider the two limiting cases of (a) a noise-free signal $SNR(f) = \infty$ and (b) an extremely noisy signal $SNR(f) = 0$. 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 $SNR(f) = 0$, $W(f) = 0$. Therefore, *for additive noise, the Wiener filter attenuates each frequency component in proportion to an estimate of the signal-to-noise ratio*.

An alternative illustration of the variations of the Wiener filter frequency response with $SNR(f)$ 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 $SNR(f)$ 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.

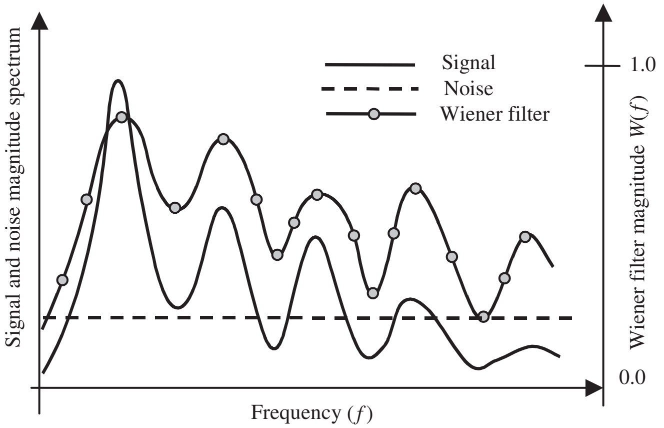


Figure 6.5 Illustration of the variation of Wiener frequency response with signal spectrum for additive white noise. The Wiener filter response broadly follows the signal spectrum.

6.6.2 Wiener Filter and Separability of Signal and Noise

In single-input noise reduction applications, where only one sensor is available (for example for speech enhancement on a mobile phone), the signal and noise cannot be perfectly separated unless their spectra are non-overlapping. A stochastic signal is completely recoverable from noise if and only 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

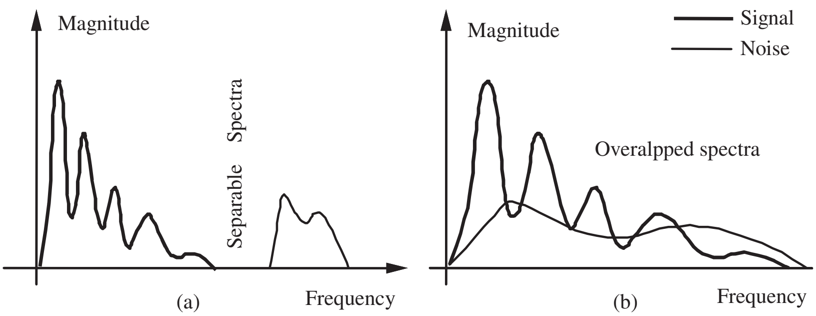


Figure 6.6 Illustration of separability: (a) the signal and noise spectra do not overlap, the signal can be recovered by a low-pass filter, (b) the signal and noise spectra overlap, the noise can be reduced but not completely removed.

noise process with overlapping spectra. For this case, it is not possible to completely separate the signal from the noise. However, 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 (6.55).

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 (6.41). Taking the expectation of the squared magnitude of both sides of Equation (6.41) yields the power spectrum of the filtered signal as

$$\begin{aligned}\mathcal{E}[|\hat{X}(f)|^2] &= |W(f)|^2\, \mathcal{E}[|Y(f)|^2] \\ &= |W(f)|^2\, P_{YY}(f)\end{aligned} \tag{6.56}$$

Substitution of $W(f)$ from Equation (6.46) in Equation (6.56) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XY}^2(f)}{P_{YY}(f)} \tag{6.57}$$

Now, for a signal observed in an uncorrelated additive noise we have

$$P_{YY}(f) = P_{XX}(f) + P_{NN}(f) \tag{6.58}$$

and

$$P_{XY}(f) = P_{XX}(f) \tag{6.59}$$

Substitution of Equations (6.58) and (6.59) in Equation (6.57) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XX}^2(f)}{P_{XX}(f) + P_{NN}(f)} \tag{6.60}$$

Now, in Equation (6.41) if instead of the Wiener filter, the square root of the Wiener filter magnitude frequency response is used, the result is

$$\hat{X}(f) = |W(f)|^{1/2}\, Y(f) \tag{6.61}$$

and the power spectrum of the signal, filtered by the square-root Wiener filter, is given by

$$\begin{aligned}\mathcal{E}[|\hat{X}(f)|^2] &= \left[|W(f)|^{1/2}\right]^2 \mathcal{E}[|Y(f)|^2] \\ &= \frac{P_{XY}(f)}{P_{YY}(f)} P_{YY}(f) \\ &= P_{XY}(f)\end{aligned} \tag{6.62}$$

Now, for uncorrelated signal and noise Equation (6.62) becomes

$$\mathcal{E}[|\hat{X}(f)|^2] = P_{XX}(f) \tag{6.63}$$

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

$$y(m) = \sum_{k=0}^{P-1} h_k\, x(m-k) + n(m) \tag{6.64}$$

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 (6.64) becomes

$$Y(f) = X(f)H(f) + N(f) \tag{6.65}$$

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 input to the equaliser is the distorted signal from the channel output, and the desired signal is the clean signal at the channel input. Using Equation (6.46) it is easy to show that the frequency domain Wiener equaliser is given by

$$W(f) = \frac{P_{XX}(f)H^*(f)}{P_{XX}(f)\,|H(f)|^2 + P_{NN}(f)} \tag{6.66}$$

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 Chapter16.

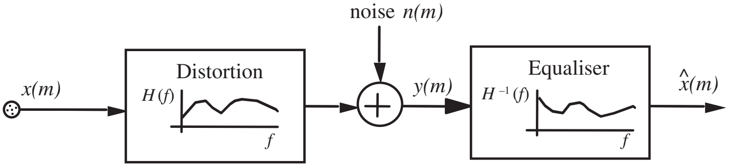


Figure 6.7 Illustration of a channel model followed by an equaliser.

6.6.5 Time-Alignment of Signals in Multi-channel/Multi-sensor Systems

In multi-channel/multi-sensor signal processing there are a 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 is 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

$$y_1(m) = x(m) + n_1(m) \tag{6.67}$$

$$y_2(m) = Ax(m-D) + n_2(m) \tag{6.68}$$

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

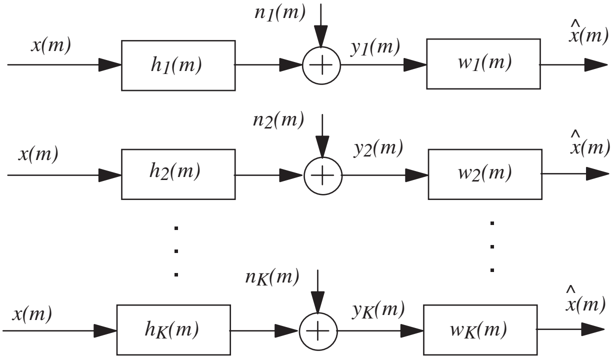


Figure 6.8 Illustration of a multi-channel system where Wiener filters are used to time-align the signals from different channels.

scaling factor. Now assume that $y_1(m)$ is used as the input 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

$$\begin{aligned} e(m) &= y_2(m) - \sum_{k=0}^{P-1} w_k y_1(m) \\ &= \left(Ax(m-D) - \sum_{k=0}^{P-1} w_k x(m) \right) + \left(\sum_{k=0}^{P-1} w_k n_1(m) \right) + n_2(m) \end{aligned} \tag{6.69}$$

The Wiener filter strives to minimise the terms shown inside the square brackets in Equation (6.69). Using the Wiener filter Equation (6.10), we have

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{R}_{y_1 y_1}^{-1} \boldsymbol{r}_{y_1 y_2} \\ &= \left(\boldsymbol{R}_{xx} + \boldsymbol{R}_{n_1 n_1} \right)^{-1} A \boldsymbol{r}_{xx}(D) \end{aligned} \tag{6.70}$$

where $\boldsymbol{r}_{xx}(D) = E\left[x(m-D)\boldsymbol{x}(m)\right]$. The frequency-domain equivalent of Equation (6.70) can be derived as

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{N_1 N_1}(f)} A e^{-j\omega D} \tag{6.71}$$

Note that in the absence of noise, the Wiener filter becomes a pure phase (or a pure delay) filter, $W(f) = Ae^{-j\omega D}$, with a flat magnitude response.

6.7 Implementation of Wiener Filters

The implementation of a Wiener filter for additive noise reduction, using Equation (6.52) or (6.54), 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 Chapter 5 and Chapter 17.

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 quasi-stationary, 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, office noise from computer machines, etc.

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 $SNR(f)$.

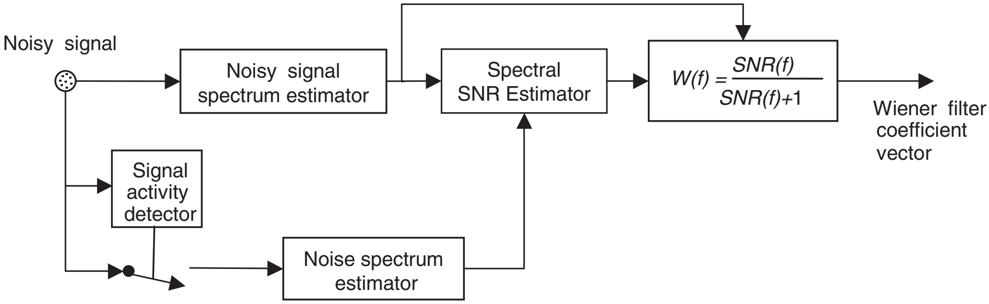

Noisy signal
Noisy signal spectrum estimator
Spectral SNR Estimator
$W(f) = \frac{SNR(f)}{SNR(f)+1}$
Wiener filter coefficient vector
Signal activity detector
Noise spectrum estimator

Figure 6.9 Configuration of a system for estimation of frequency Wiener filter.

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 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 Wiener filter solution; a large filter may produce an ill-conditioned large-dimensional correlation matrix in Equation (6.10).

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 and 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

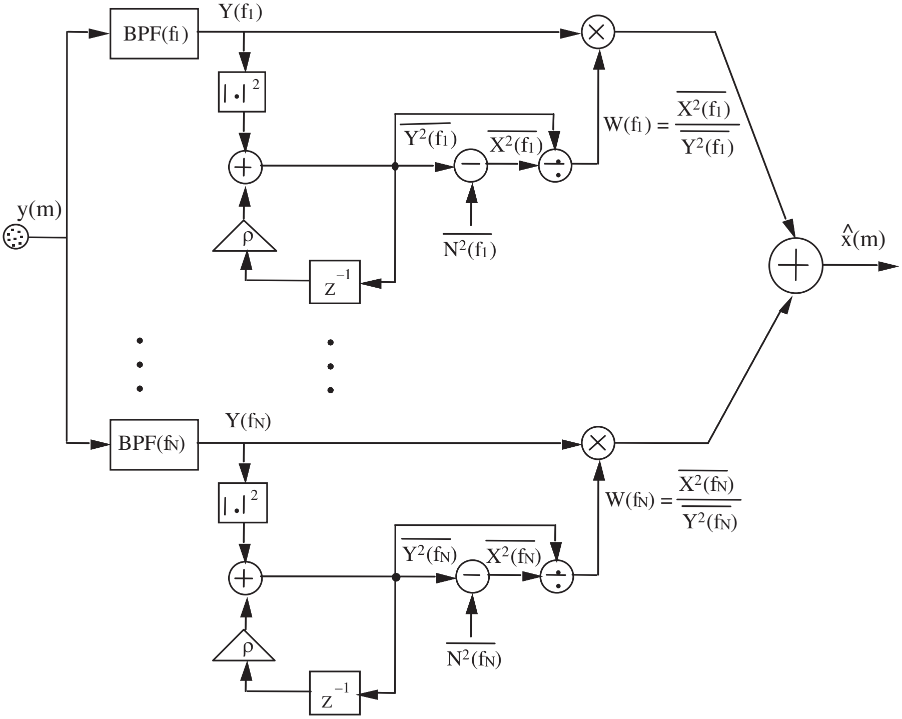


Figure 6.10 A filter-bank implementation of a Wiener filter for additive noise reduction.

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 15, 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 non-stationarity: 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 non-linear distortion in the input will affect the filter performance.

The performance of 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 non-linear distortions of the input signal, the minimum mean square error would be non-zero. 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. Automatic Control*, **AC-19**: 716–723.

Alexander S.T. (1986) *Adaptive Signal Processing Theory and Applications*. Springer-Verlag, 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*. John Wiley & Sons, Inc, New York.

Durbin J. (1959) Efficient Estimation of Parameters in Moving Average Models. *Biometrica*, **46**: 306–316.

Giordano A.A. and Hsu F.M. (1985) *Least Square Estimation with Applications to Digital Signal Processing*. John Wiley & Sons, Inc, 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. *Numerical Mathematics*, **14**: 403–420.

Golub G.H. and Van Loan C.F. (1980) An Analysis of the Total Least Squares Problem. *SIAM J. of Numerical Analysis*, **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-Verlag, 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. Info. Theory*, **IT-20**: 146–181.

Kailath T. (1977) *Linear Least Squares Estimation, Benchmark Papers in Electrical Engineering and Computer Science*. Dowden, Hutchinson & Ross.

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. Automatic Control*, **AC-25**: 164–176.

Kolmogrov A.N. (1939) Sur l' Interpolation et Extrapolation des Suites Stationaires. *Comptes Rendus de l'Academie des Sciences*, **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.

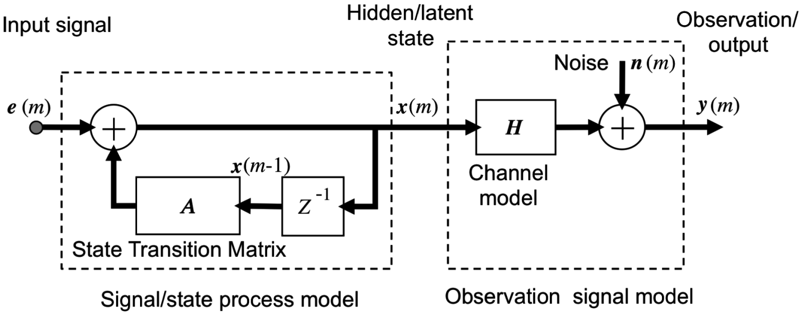
Strang G. (1976) *Linear Algebra and Its Applications*, 3rd edn, Harcourt Brace Jovanovich, San Diego, CA.

Whittle P.W. (1983) *Prediction and Regulation by Linear Least-Squares Methods*. University of Minnesota Press, Minneapolis, Minnesota.

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.

Wold H. (1954) *The Analysis of Stationary Time Series*, 2nd edn, Almquist and Wicksell, Uppsala.



7

Adaptive Filters: Kalman, RLS, LMS

Adaptive filters are a class of recursive signal or parameter estimation methods that appear in many signal processing and communication systems for applications such as system identification, channel equalisation, echo cancellation, noise reduction, radar/sonar signal processing, the tracking of the position and velocity of objects or vehicles e.g. in GPS systems, computer vision, beam-forming, space-time signal processing in mobile communication audio coding and video coding.

Adaptive filters work on the principle of estimating a noisy signal or hidden parameters by minimising an objective error function, usually the mean squared difference (or error) between the filter output signal and a target (or desired) signal. Adaptive filters are used for estimation and identification of non-stationary signals, channels and systems or in applications where a sample-by-sample adaptation of a process and/or a low processing delay is required.

This chapter begins with an introduction to adaptive filtering methods. Next, we study the theory of state-space Kalman filters. 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 that degrade the clean signal. Although Kalman filter theory assumes that the statistical parameters of the signal, noise and channel are known and available for use, in practice these parameters are unknown and need to be estimated and updated at each recursion. We will also study the extension of linear Kalman filter to nonlinear transition and observation functions; namely the extended Kalman filter (EKF) and the unscented Kalman filter ((UKF).

Next 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. For 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 and its energy-normalised version.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

7.1 Introduction

Adaptive filters are used in applications that involve a combination of three broad signal processing problems:

(1) De-noising and channel equalisation: for example the adaptive filtering of a non-stationary distorted and/or noisy signal, such as speech or image, to remove the effects of noise and channel distortions.
(2) Trajectory estimation: tracking and prediction of the trajectory of a non-stationary object, signal or parameter observed in noise, such as a moving car with GPS which may go through tunnels with no GPS signal coverage. Kalman filters were also used in the Apollo space programme for trajectory estimation.
(3) System identification: adaptive estimation of the parameters of an unknown time-varying system from related observations, for example in the acoustic modelling of the impulse response of a room or a musical instrument or the modeling of an echo path for echo cancellation.

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 minimising an objective function; often the mean squared error between the filter output and a target (or desired) signal. The use of the least squared error (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 the estimation error at time m is orthogonal to all the information available up to and including time m. In Chapter 18, on independent component analysis, adaptive filters with non-linear objective functions are considered.

An adaptive filter can be a combination of the following types of filters:

- single input or multiple-inputs filters;
- scalar-valued input or vector-valued input vectors;
- linear or non-linear 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 types of adaptive filters. The adaptation algorithm can be based on a variant of one of the three most commonly used adaptive estimation methods, namely

- state-space Kalman filters;
- recursive least squared (RLS) filters;
- least mean squared (LMS) filters.

Table 7.1 Comparison of different adaptive/recursive filtering methods.

Method	Operation/ Adaptation	Input–output model	Statistics required	Gain vector
Kalman	Sample recursive	State-space models	Signal and noise covariance matrices, system and channel transition matrices	Kalman gain
LS/Wiener	Block adaptive	Filter	Correlation matrices	None
RLS	Recursive LS/Wiener	Filter	Recursive calculation of correlation matrices	RLS gain
LMS	Gradient sample adaptive	Filter	None	LMS gain

The different types of adaptation algorithms differ mainly in terms of the use of prior or estimated knowledge of the system function and the covariance matrices of signal and noise and also in terms of the complexity of the solutions, see Table 7.1.

7.2 State-Space Kalman Filters

Kalman filter, illustrated in Figure 7.1, is a Bayesian recursive least square error method for estimation of a signal or a parameter vector distorted in transmission through a channel and observed in noise. Kalman filters can be used with time-varying as well as time-invariant processes.

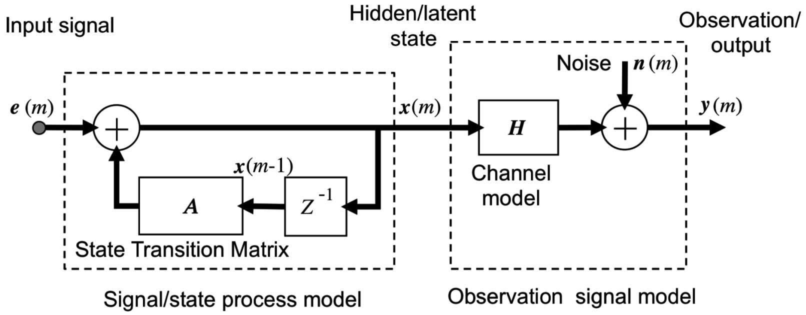


Figure 7.1 Illustration of signal (or state) and observation models in Kalman filter.

The Kalman filter is named after Rudolf E. Kalman who published his work in 1960. However, similar filters were formulated by Thorvald Nicolai Thiele in 1880 and Peter Swerling in 1958. The Kalman filter is also a special case of a more general non-linear filter developed by Ruslan Stratonovich and published in 1957.

The Kalman filter was applied for trajectory estimation in the Apollo space programme; it has many applications, for example it is widely used in process control, in radio and other communication devices particularly as phase-lock-loop systems, in GPS position tracking and guidance systems and in signal denoising and system identification problems.

The Kalman filter is a Bayesian filter in that it employs the prior probability distributions of the signal and noise processes, the signal is assumed to be a zero-mean Gaussian-Markov process whereas the noise is assumed to be zero-mean independent identically distributed (IID) Gaussian process. The filter also assumes that the parameters of the models of signal and noise generation and channel distortion are known a priori.

Kalman filter formulation 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 and distorted observation signal. For a signal vector $\boldsymbol{x}(m)$ and noisy observation vector $\boldsymbol{y}(m)$, Equations describing the state process model and the observation model are defined as

$$\boldsymbol{x}(m) = \boldsymbol{A}\,\boldsymbol{x}(m-1) + \boldsymbol{B}\,\boldsymbol{u}(m) + \boldsymbol{e}(m) \tag{7.1}$$

$$\boldsymbol{y}(m) = \boldsymbol{H}\,\boldsymbol{x}(m) + \boldsymbol{n}(m) \tag{7.2}$$

where

$\boldsymbol{x}(m)$ is the P-dimensional signal, or the state parameter vector at time m;
$\boldsymbol{A}$ is a $P \times P$ dimensional state transition matrix that relates the states of the process at times m and $m-1$;

$\boldsymbol{B}$ is a $P \times P$ dimensional control matrix, used in process control modelling;
$\boldsymbol{u}(m)$ is the P-dimensional control input;
$\boldsymbol{e}(m)$ is the P-dimensional uncorrelated input excitation vector of the state Equation, $\boldsymbol{e}(m)$ is a normal (Gaussian) process;
$\boldsymbol{n}(m)$ is an M-dimensional noise vector, also known as measurement noise, $\boldsymbol{n}(m)$ is a normal (Gaussian) process $p\big(\boldsymbol{n}(m)\big) \sim N(0, \boldsymbol{R})$;
$\boldsymbol{y}(m)$ is the M-dimensional noisy and distorted observation vector;
$\boldsymbol{H}$ is the $M \times P$ dimensional channel distortion matrix;
$\boldsymbol{Q}$ is the $P \times P$ dimensional covariance matrix of $\boldsymbol{e}(m)$;
$\boldsymbol{R}$ is the $M \times M$ dimensional covariance matrix of $\boldsymbol{n}(m)$.

Note that the control matrix $\boldsymbol{B}$ and the control vector $\boldsymbol{u}(m)$ are used in control applications where often an external input may be applied by a controller process to change, adjust or correct the trajectory of the vector process $\boldsymbol{x}(m)$.

In communication signal processing applications, such as channel equalisation or speech enhancement, often there is no external control input vector $\boldsymbol{u}(m)$ and the Kalman Equations reduce to

$$\boldsymbol{x}(m) = \boldsymbol{A}\,\boldsymbol{x}(m-1) + \boldsymbol{e}(m) \tag{7.3}$$

$$\boldsymbol{y}(m) = \boldsymbol{H}\,\boldsymbol{x}(m) + \boldsymbol{n}(m) \tag{7.4}$$

The Kalman filter algorithm is described.

Box 1 – Kalman Filter Algorithm

Input: observation vectors $\{\boldsymbol{y}(m)\}$
Output: estimates of state or signal vectors $\{\hat{\boldsymbol{x}}(m)\}$

Initial Conditions
Prediction error covariance matrix:

$$\boldsymbol{P}(0|-1) = \delta\mathbf{I} \tag{7.5}$$

Prediction:

$$\hat{\boldsymbol{x}}(0|-1) = 0 \tag{7.6}$$

For $m = 0,\ 1,\ \ldots$

Time-Update, Process Prediction Equations
State vector prediction:

$$\hat{\boldsymbol{x}}(m|m-1) = \boldsymbol{A}\,\hat{\boldsymbol{x}}(m-1) \tag{7.7}$$

Covariance matrix of prediction error:

$$\boldsymbol{P}(m|m-1) = A\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{7.8}$$

Measurement-Update, Estimation Equations
Kalman gain vector:

$$\boldsymbol{K}(m) = \boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}\big(H\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\big)^{-1} \tag{7.9}$$

State vector estimation:

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\big(\boldsymbol{y}(m) - \boldsymbol{H}\hat{\boldsymbol{x}}(m|m-1)\big) \tag{7.10}$$

Covariance matrix of estimation error:

$$\boldsymbol{P}(m) = [\boldsymbol{I} - \boldsymbol{K}H]\boldsymbol{P}(m|m-1) \tag{7.11}$$

7.2.1 Derivation of Kalman Filter Algorithm

Kalman filter can be derived as a recursive minimum mean square error estimator of a signal (or state vector) $\boldsymbol{x}(m)$ from a noisy observation $\boldsymbol{y}(m)$. The derivation of Kalman filter assumes that the state transition matrix $\boldsymbol{A}$, the channel distortion matrix $\boldsymbol{H}$, the covariance matrix $\boldsymbol{Q}$ of the input $\boldsymbol{e}(m)$ of the state Equation and the covariance matrix $\boldsymbol{R}$ of the additive noise $\boldsymbol{n}(m)$ are given or estimated by some means.

The derivation of Kalman filter, described next, is based on the following recursive methodology where each recursion consists of two stages as shown in Figure 7.2:

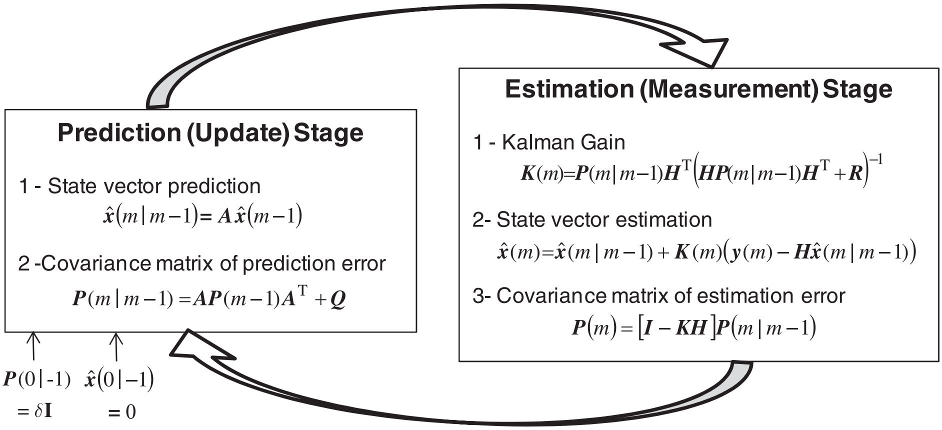

Estimation (Measurement) Stage
1 - Kalman Gain
$\boldsymbol{K}(m)=\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}\right)^{-1}$
2- State vector estimation
$\hat{\boldsymbol{x}}(m)=\hat{\boldsymbol{x}}(m|m-1)+\boldsymbol{K}(m)\left(\boldsymbol{y}(m)-\boldsymbol{H}\hat{\boldsymbol{x}}(m|m-1)\right)$
3- Covariance matrix of estimation error
$\boldsymbol{P}(m)=\left[\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H}\right]\boldsymbol{P}(m|m-1)$
Prediction (Update) Stage
1 - State vector prediction
$\hat{\boldsymbol{x}}(m|m-1)=\boldsymbol{A}\hat{\boldsymbol{x}}(m-1)$
2 -Covariance matrix of prediction error
$\boldsymbol{P}(m|m-1)=\boldsymbol{A}\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{Q}$
$\boldsymbol{P}(0|-1)$ $=\delta\mathbf{I}$
$\hat{\boldsymbol{x}}(0|-1)$ $=0$

Figure 7.2 Illustration of the prediction-estimation stages at each iteration/update of a Kalman filter.

(1) Prediction (update) stage: the signal state is predicted from the previous observations and a prediction error covariance matrix is obtained as expressed in Equations (7.7)–(7.8).

(2) Estimation (measurement) stage: the results of prediction from step (1) and the innovation signal (note innovation is the difference between prediction and noisy observation) are used to estimate the signal. At this stage the Kalman gain vector and estimation error covariance matrix are calculated as expressed in Equations (7.9)–(7.11).

In this chapter, we use the notation $\hat{\boldsymbol{y}}\,(m\,|m-i\,)$ to denote a prediction of $\boldsymbol{y}(m)$ based on the observation samples up to the time $m-i$. Assume that $\hat{\boldsymbol{x}}\,(m|m-1)$ is the least square error prediction of $\boldsymbol{x}(m)$ based on the observations $[\boldsymbol{y}(0),\ldots\boldsymbol{y}(m-1)]$. Define a prediction Equation as

$$\hat{\boldsymbol{x}}(m|m-1) = \boldsymbol{A}\hat{\boldsymbol{x}}(m-1) \tag{7.12}$$

An innovation signal, composed of prediction error plus noise, may be defined as

$$\boldsymbol{v}(m) = \boldsymbol{y}(m) - \boldsymbol{H}\hat{\boldsymbol{x}}(m|m-1) \tag{7.13}$$

where $\hat{\boldsymbol{x}}\,(m|m-1)$ denotes the least square error prediction of the signal $\boldsymbol{x}(m)$. The innovation signal vector $\boldsymbol{v}(m)$ is the part of the observation signal $\boldsymbol{y}(m)$ that is unpredictable from the past observations; it includes both the noise and the information-bearing unpredictable part of the signal $\boldsymbol{x}(m)$.

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

$$\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{y}^{\mathrm{T}}(m-k)\right] = 0 \quad k > 0 \tag{7.14}$$

and

$$\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(k)\right] = 0, \quad m \neq k \tag{7.15}$$

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 (7.14) is used as the starting point to derive an optimal linear filter whose innovations $\boldsymbol{v}(m)$ are orthogonal to the past observations $\boldsymbol{y}(m)$.

Substituting the observation Equation (7.2) in Equation (7.13) yields

$$\begin{aligned} \boldsymbol{v}(m) &= \boldsymbol{H}\,\boldsymbol{x}(m) + \boldsymbol{n}(m) - \boldsymbol{H}\,\hat{\boldsymbol{x}}\,(m\,|m-1\,) \\ &= \boldsymbol{H}\big(\boldsymbol{x}(m) - \hat{\boldsymbol{x}}(m|m-1)\big) + \boldsymbol{n}(m) \\ &= \boldsymbol{H}\,\tilde{\boldsymbol{x}}(m|m-1) + \boldsymbol{n}(m) \end{aligned} \tag{7.16}$$

where $\tilde{\boldsymbol{x}}(m|m-1)$ is the signal prediction error vector defined as

$$\tilde{\boldsymbol{x}}(m|m-1) = \boldsymbol{x}(m) - \hat{\boldsymbol{x}}\,(m\,|m-1\,) \tag{7.17}$$

From Equation (7.16) the covariance matrix of the innovation signal $\boldsymbol{v}(m)$ is given by

$$\mathcal{E}[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(m)] = \boldsymbol{H}\,\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R} \tag{7.18}$$

where $\boldsymbol{P}(m|m-1)$ is the covariance matrix of the prediction error $\tilde{\boldsymbol{x}}(m|m-1)$. The estimation of $\boldsymbol{X}(m)$, based on the samples available up to the time m, can be expressed recursively as a linear combination of the prediction of $\boldsymbol{x}(m)$ based on the samples available up to the time $m-1$ and the innovation signal at time m as

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.19}$$

where the $P \times M$ matrix $\boldsymbol{K}$(m) is the Kalman gain matrix. Now, from Equation (7.1), we have

$$\hat{\boldsymbol{x}}(m|m-1) = \boldsymbol{A}\,\hat{\boldsymbol{x}}(m-1) \tag{7.20}$$

Substitution of Equation (7.20) in (7.19) gives

$$\hat{\boldsymbol{x}}(m) = \boldsymbol{A}\hat{\boldsymbol{x}}(m-1) + \boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.21}$$

To obtain a recursive relation for the computation and update of the Kalman gain matrix, multiply both sides of Equation (7.19) by $\boldsymbol{v}^{\mathrm{T}}(m)$ and take the expectation of the results to yield

$$\mathcal{E}[\hat{\boldsymbol{x}}(m)\boldsymbol{v}^{\mathrm{T}}(m)] = \mathcal{E}[\hat{\boldsymbol{x}}(m|m-1)\boldsymbol{v}^{\mathrm{T}}(m)] + \boldsymbol{K}(m)\mathcal{E}[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(m)] \tag{7.22}$$

Owing to the required orthogonality of the innovation sequence $\boldsymbol{v}(m)$ to the past samples, we have

$$\mathcal{E}[\hat{\boldsymbol{x}}(m|m-1)\boldsymbol{v}^{\mathrm{T}}(m)] = 0 \tag{7.23}$$

Hence, from Equations (7.22) and (7.23), the Kalman gain matrix is given by

$$\boldsymbol{K}(m) = \mathcal{E}[\hat{\boldsymbol{x}}(m)\boldsymbol{v}^{\mathrm{T}}(m)]\mathcal{E}[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(m)]^{-1} \tag{7.24}$$

The first term on the right-hand side of Equation (7.24) can be expressed as

$$\begin{aligned}
\mathcal{E}[\hat{\boldsymbol{x}}(m)\boldsymbol{v}^{\mathrm{T}}(m)] &= \mathcal{E}[(\boldsymbol{x}(m) - \tilde{\boldsymbol{x}}(m))\boldsymbol{v}^{\mathrm{T}}(m)] \\
&= \mathcal{E}[(\hat{\boldsymbol{x}}(m|m-1) + \tilde{\boldsymbol{x}}(m|m-1) - \tilde{\boldsymbol{x}}(m))\boldsymbol{v}^{\mathrm{T}}(m)] \\
&= \mathcal{E}[\tilde{\boldsymbol{x}}(m|m-1)\boldsymbol{v}^{\mathrm{T}}(m)] \\
&= \mathcal{E}[\tilde{\boldsymbol{x}}(m|m-1)(\boldsymbol{H}\,\boldsymbol{x}(m) + \boldsymbol{n}(m) - \boldsymbol{H}\hat{\boldsymbol{x}}(m|m-1))^{\mathrm{T}}] \\
&= \mathcal{E}[\tilde{\boldsymbol{x}}(m|m-1)\tilde{\boldsymbol{x}}^{\mathrm{T}}(m|m-1)\boldsymbol{H}^{\mathrm{T}} \\
&= \boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}
\end{aligned} \tag{7.25}$$

where $\tilde{\boldsymbol{x}}(m)$ is the estimation error vector. In developing the successive lines of Equation (7.25), we have used the following relations:

$$\boldsymbol{x}(m) = \hat{\boldsymbol{x}}(m|m-1) + \tilde{\boldsymbol{x}}\,(m|m-1) \tag{7.26}$$

$$\mathcal{E}\big[\tilde{\boldsymbol{x}}\,(m)\,\boldsymbol{v}^{\mathrm{T}}(m)\big] = 0 \tag{7.27}$$

$$\mathcal{E}\big[\hat{\boldsymbol{x}}\big(m|m-1\big)\boldsymbol{v}^{\mathrm{T}}(m)\big] = 0 \tag{7.28}$$

$$\mathcal{E}\big[\tilde{\boldsymbol{x}}\,(m|m-1)\,\boldsymbol{n}^{\mathrm{T}}(m)\big] = 0 \tag{7.29}$$

Substitution of Equations (7.18) and (7.25) in Equation (7.24) yields the following Equation for the Kalman gain matrix:

$$\boldsymbol{K}\,(m) = \boldsymbol{P}\,(m|m-1)\,\boldsymbol{H}^{\mathrm{T}}\big[\boldsymbol{H}\boldsymbol{P}\,(m|m-1)\,\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}\big]^{-1} \tag{7.30}$$

where $\boldsymbol{P}(m|m-1)$ is the covariance matrix of the signal prediction error $\tilde{\boldsymbol{x}}(m|m-1)$. Note that the Kalman gain vector can be interpreted as a function of signal-to-noise ratio of the innovation signal; it performs on innovation a function similar to Wiener filtering.

A recursive relation for calculation of $\boldsymbol{P}(m|m-1)$, the covariance matrix of prediction error $\tilde{\boldsymbol{x}}\,(m\,|m-1\,)$ is derived as follows:

$$\tilde{\boldsymbol{x}}\,(m\,|m-1\,) = \boldsymbol{x}\,(m) - \hat{\boldsymbol{x}}\,(m\,|m-1\,) \tag{7.31}$$

Substitution of Equation (7.1) and (7.20) in Equation (7.31) and rearrangement of the terms yields

$$\begin{aligned}
\tilde{\boldsymbol{x}}\,(m|m-1) &= [\boldsymbol{A}\,\boldsymbol{x}(m-1) + \boldsymbol{e}(m)] - \big[\boldsymbol{A}\,\hat{\boldsymbol{x}}\,(m-1)\big] \\
&= \boldsymbol{A}\,\tilde{\boldsymbol{x}}\,(m-1) + \boldsymbol{e}(m)
\end{aligned} \tag{7.32}$$

The covariance matrix of $\tilde{\boldsymbol{x}}\,(m\,|m-1\,)$ is obtained as

$$\mathcal{E}\big[\tilde{\boldsymbol{x}}\,(m|m-1)\,\tilde{\boldsymbol{x}}\,(m|m-1)^{\mathrm{T}}\big] = \boldsymbol{A}\mathcal{E}\big[\tilde{\boldsymbol{x}}\,(m-1)\,\tilde{\boldsymbol{x}}\,(m-1)^{\mathrm{T}}\big]\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{7.33}$$

or

$$\boldsymbol{P}(m|m-1) = \boldsymbol{A}\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{7.34}$$

where $\boldsymbol{P}(m|m-1)$ and $\boldsymbol{P}(m)$ are the covariance matrices of the prediction error $\tilde{\boldsymbol{x}}\,(m\,|m-1\,)$ and estimation error $\tilde{\boldsymbol{x}}\,(m)$ respectively. A recursive relation for the covariance matrix of the signal estimation error vector $\tilde{\boldsymbol{x}}\,(m)$ can be derived as follows. Subtracting both sides of Equation (7.19) from $\boldsymbol{x}(m)$ we have

$$\tilde{\boldsymbol{x}}(m) = \tilde{\boldsymbol{x}}(m|m-1) - \boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.35}$$

From Equation (7.35) the covariance matrix of the estimation error vector can be expressed as

$$\begin{aligned}\mathcal{E}\left[\tilde{\boldsymbol{x}}(m)\,\tilde{\boldsymbol{x}}(m)^T\right] &= \mathcal{E}\left[\tilde{\boldsymbol{x}}(m|m-1)\,\tilde{\boldsymbol{x}}(m|m-1)^T\right] + \boldsymbol{K}(m)\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}(m)^T\right]\,\boldsymbol{K}(m)^T \\ &\quad - 2\mathcal{E}\left[\tilde{\boldsymbol{x}}(m|m-1)\boldsymbol{v}(m)^T\right]\boldsymbol{K}(m)^T \end{aligned} \tag{7.36}$$

From Equations (7.24) and (7.25) we have

$$\boldsymbol{K}(m)\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}(m)^{\mathrm{T}}\right] = \boldsymbol{P}(m|m-1)\boldsymbol{H}^T \tag{7.37}$$

From Equation (7.25) we have

$$\mathcal{E}\left[\tilde{\boldsymbol{x}}(m|m-1)\boldsymbol{v}(m)^{\mathrm{T}}\right] = \boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}} \tag{7.38}$$

Substitution of Equations (7.37) and (7.38) in (7.36) and rearranging yields

$$\boldsymbol{P}(m) = [\boldsymbol{I} - \boldsymbol{KH}]\boldsymbol{P}(m|m-1) \tag{7.39}$$

7.2.2 Recursive Bayesian Formulation of Kalman Filter

It is instructive to derive the Kalman filter using the Bayesian estimation method which serves to show that Kalman filter is a maximum a posteriori (MAP) solution.

The posterior pdf of the state vector $\boldsymbol{x}(m)$ given the observation vector $\boldsymbol{y}(m)$ can be expressed using the Baye's rule as

$$\begin{aligned} p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m),\boldsymbol{y}(m-1)\big) &= \frac{p\big(\boldsymbol{x}(m),\boldsymbol{y}(m),\boldsymbol{y}(m-1)\big)}{p\big(\boldsymbol{y}(m),\boldsymbol{y}(m-1)\big)} \\ &= \frac{p\big(\boldsymbol{y}(m)|\boldsymbol{x}(m),\boldsymbol{y}(m-1)\big)p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m-1)\big)p\big(\boldsymbol{y}(m-1)\big)}{p\big(\boldsymbol{y}(m)|\boldsymbol{y}(m-1)\big)p\big(\boldsymbol{y}(m-1)\big)} \end{aligned} \tag{7.40}$$

Since the observation $\boldsymbol{y}(m)$ depends only on the signal $\boldsymbol{x}(m)$ and noise, $p(\boldsymbol{y}(m)|\boldsymbol{x}(m),\boldsymbol{y}(m-1)) = p(\boldsymbol{y}(m)|\boldsymbol{x}(m))$, similarly, $p(\boldsymbol{x}(m)|\boldsymbol{y}(m),\boldsymbol{y}(m-1)) = p(\boldsymbol{x}(m)|\boldsymbol{y}(m))$. Hence Equation (7.40) becomes

$$\underbrace{p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m)\big)}_{a\ posteriori\ pdf} = \frac{\overbrace{p\big(\boldsymbol{y}(m)|\boldsymbol{x}(m)\big)}^{likelihood\ pdf}\ \overbrace{p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m-1)\big)}^{a\ priori\ pdf}}{\underbrace{p\big(\boldsymbol{y}(m)|\boldsymbol{y}(m-1)\big)}_{Normalisation}} \tag{7.41}$$

The l.h.s. of Equation (7.41), $p(\boldsymbol{x}(m)|\boldsymbol{y}(m))$, is the a posterior pdf and its r.h.s. is composed of the following terms:

Likelihood pdf – $p\big(\boldsymbol{y}(m)|\boldsymbol{x}(m)\big) = N(\boldsymbol{y}(m), \boldsymbol{Hx}(m), \boldsymbol{\Sigma}_{vv})$ is modelled by a Gaussian process with a mean vector of $\boldsymbol{Hx}(m)$ and a covariance matrix $\boldsymbol{\Sigma}_{vv}$ equal to that of the noise.

Prior pdf – $p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m-1)\big) = N(\boldsymbol{x}(m), \boldsymbol{A}\hat{\boldsymbol{x}}(m-1)$ is modelled by a Gaussian process with a mean vector of $\boldsymbol{A}\hat{\boldsymbol{x}}(m-1)$ and a covariance matrix $\boldsymbol{P}(m|m-1)$ equal to that of prediction error $cov(\boldsymbol{x}(m) - \boldsymbol{A}\hat{\boldsymbol{x}}(m-1))$.

Normalising term – The denominator acts as a normalising term since it can be expressed as

$$p\big(\boldsymbol{y}(m)|\boldsymbol{y}(m-1)\big) = \int p\big(\boldsymbol{y}(m)|\boldsymbol{x}(m)\big)p(\boldsymbol{x}(m)|\boldsymbol{y}(m-1))d\boldsymbol{x}(m) \tag{7.42}$$

Hence substituting the Gaussian forms of the likelihood and prior pdfs in Equation (7.41) yields

$$\begin{aligned} p\big(\boldsymbol{x}(m)|\boldsymbol{y}(m)\big) &= C\ exp\Big(-0.5\big(\boldsymbol{y}(m) - \boldsymbol{Hx}(m)\big)^T\boldsymbol{R}^{-1}\big(\boldsymbol{y}(m) - \boldsymbol{Hx}(m)\big)\Big) \\ &\quad \times exp\Big(-0.5\big(\boldsymbol{x}(m) - \boldsymbol{A}\hat{\boldsymbol{x}}(m-1)\big)^T\boldsymbol{P}(m|m-1)^{-1}\big(\boldsymbol{x}(m) - \boldsymbol{A}\hat{\boldsymbol{x}}(m-1)\big)\Big) \end{aligned} \tag{7.43}$$

where $C = \dfrac{1}{p(\boldsymbol{y}(m)|\boldsymbol{y}(m-1))} \dfrac{1}{(2\pi)^P} |\boldsymbol{R}|^{-1} |\boldsymbol{P}(m|m-1)|^{-1}$. The MAP solution is obtained from the point where the gradient of the a posteriori pdf is zero as

$$\hat{\boldsymbol{x}}_{MAP}(m) = \underset{\boldsymbol{x}(m)}{\arg\ zero} \left(\frac{\partial \log p(\boldsymbol{x}(m)|\boldsymbol{y}(m))}{\partial \boldsymbol{x}(m)} \right) \tag{7.44}$$

From Equations (7.43) and (7.44) the MAP solution is given by

$$\hat{\boldsymbol{x}}_{MAP}(m) = \left(\boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H} + \boldsymbol{P}^{-1}(m|m-1)\right)^{-1} \times \left(\boldsymbol{P}^{-1}(m|m-1)\boldsymbol{A}\hat{\boldsymbol{x}}(m-1) + \boldsymbol{H}\boldsymbol{R}^{-1}\boldsymbol{y}(m)\right) \tag{7.45}$$

Using the matrix inverse lemma (proved in Section 7.4) which states that if a matrix $\boldsymbol{A} = \boldsymbol{B}^{-1} + \boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^T$, it follows that $\boldsymbol{A}^{-1} = \boldsymbol{B} - \boldsymbol{B}\boldsymbol{C}(\boldsymbol{D} + \boldsymbol{C}^T\ \boldsymbol{B}\boldsymbol{C})^{-1}\boldsymbol{C}^T\ \boldsymbol{B}$, and substituting $\boldsymbol{B} = \boldsymbol{P}(m|m-1)$, $\boldsymbol{C} = \boldsymbol{H}^T$ and $\boldsymbol{D} = \boldsymbol{R}$, Equation can be rewritten as

$$\hat{\boldsymbol{x}}_{MAP}(m) = \boldsymbol{A}\hat{\boldsymbol{x}}(m-1) + \boldsymbol{K}(m)\left(\boldsymbol{y}(m) - \boldsymbol{H}\boldsymbol{A}\hat{\boldsymbol{x}}(m-1)\right) \tag{7.46}$$

where $\boldsymbol{K}(m)$ is the Kalman gain is given by

$$\boldsymbol{K}(m) = \boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R})^{-1} \tag{7.47}$$

Hence Kalman filter is a Bayesian MAP estimator.

7.2.3 Markovian Property of Kalman Filter

The Kalman filter model for the signal process is a Markovian model in that the state vector at time m, $\boldsymbol{x}(m)$, depends only on the state vector at time $m-1$, $\boldsymbol{x}(m-1)$, and is independent of the previous states as

$$p(\boldsymbol{x}(m)|[\boldsymbol{x}(m-1) \cdots \boldsymbol{x}(0)]) = p(\boldsymbol{x}(m)|\boldsymbol{x}(m-1)) \tag{7.48}$$

The pdf $p(\boldsymbol{x}(m)|\boldsymbol{x}(m-1))$ acts as a predictive prior pdf, it is given by

$$p\left(\boldsymbol{x}(m)|\boldsymbol{x}(m-1)\right) = N\left(\boldsymbol{x}(m), \boldsymbol{A}\boldsymbol{x}(m-1), \boldsymbol{Q}\right) \tag{7.49}$$

where $N(\boldsymbol{x}(m), \boldsymbol{A}\boldsymbol{x}(m-1), \boldsymbol{Q})$ denotes a normal (i.e. Gaussian) process, with a mean vector of $\boldsymbol{A}\boldsymbol{x}(m-1)$ and a covariance matrix of $\boldsymbol{Q}$, this pdf is given by

$$N\left(\boldsymbol{x}(m), \boldsymbol{A}\boldsymbol{x}(m-1), \boldsymbol{Q}\right) = \frac{1}{(2\pi)^{P/2}} |\boldsymbol{Q}|^{-1} \exp\left(-0.5\left(\boldsymbol{x}(m) - \boldsymbol{A}\boldsymbol{x}(m-1)\right)^{\mathrm{T}} \boldsymbol{Q}^{-1} \left(\boldsymbol{x}(m) - \boldsymbol{A}\boldsymbol{x}(m-1)\right)\right) \tag{7.50}$$

Equation (7.50) is a probabilistic description of the signal generation process model in Kalman filter as described in Equation (7.3). From the Markovian property, expressed in Equation (7.48), the probability of a time series of N samples $[\boldsymbol{x}(0) \cdots \boldsymbol{x}(N-1)]$ can be expressed as

$$p\left([\boldsymbol{x}(0), \cdots, \boldsymbol{x}(N-1)]\right) = p\left(\boldsymbol{x}(0)\right) \prod_{m=1}^{N-1} p\left(\boldsymbol{x}(m)|\boldsymbol{x}(m-1)\right) = p\left(\boldsymbol{x}(0)\right) \prod_{m=1}^{N-1} N\left(\boldsymbol{x}(m), \boldsymbol{A}\boldsymbol{x}(m-1), \boldsymbol{Q}\right) \tag{7.51}$$

Note that the above Equations model the signal generation process. The process is observed in noise and the Kalman filter provides a MAP solution as described in the previous section.

7.2.4 Comparison of Kalman filter and hidden Markov model

Kalman filter and hidden Markov model (HMM) have the following commonalities and differences:

(1) Markovian processes. Both Kalman filter and HMM are first order Markovian processes, this implies that the current state of the process depends on its previous state only.
(2) Finite-state versus continuous-state variables. An HMM has a finite-number of states whereas a Kalman filter has continuously-variable states.
(3) Evolution of the Markovian states. In HMM the state process switch according to a Markovian probability described by the state transition matrix whereas in Kalman filter, the state evolves according to a state transition matrix and a Gaussian process noise.
(4) Gaussian observations. In both HMMs and Kalman filters the observed signals are Gaussian distributed.
(5) Hidden states. In HMMs the hidden (latent) states generate the observed signals, i.e. the states are hidden even when there is no channel distortion or noise. In contrast in Kalman filter the states are unobservable because of noise and distortion.

7.2.5 Comparison of Kalman and Wiener Filters

The main differences and similarities between the Wiener and Kalman filters are as follows.

(1) Single Input versus Multiple Inputs. Wiener filter is formulated for estimation of a scalar signal or the identification of the parameter vector of a filter with scalar inputs/outputs. In contrast Kalman filter can be used for multiple input multiple output (MIMO) scenarios.
(2) Dynamic Model in Kalman versus Stationary Model in Wiener. The Kalman filter Equations include a dynamic model of the state vector and a model of the noisy observation, whereas the Wiener filter Equation includes the cross correlations of the clean signal and the observation and the autocorrelation of the observation. For stationary signals Wiener and Kalman produce the same results.
(3) Mean squared error versus Bayesian solution. The Wiener filter is a maximum likelihood solution for stationary Gaussian processes whereas the Kalman filter is a Bayesian MAP solution for Gaussian-Markovian processes.
(4) The Kalman gain Equation is similar to the Wiener filter Equation.

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

$$x(m) = x(m-1) = x \tag{7.52}$$

$$y(m) = x + n(m) \tag{7.53}$$

Note that for this scalar process the state transition parameter $a = 1$, the state excitation signal $e(m) = 0$, 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

$$P(-1) = \delta \tag{7.54}$$

$$\hat{x}(0|-1) = 0 \tag{7.55}$$

For $m = 0,\ 1,\ \ldots$

Time-Update Equations
Signal prediction Equation:

$$\hat{x}(m|m-1) = \hat{x}(m-1) \tag{7.56}$$

Covariance matrix of prediction error:

$$P(m|m-1) = P(m-1) \tag{7.57}$$

Measurement-Update Equations
Kalman gain vector:

$$K(m) = P(m|m-1)(P(m|m-1) + \sigma_n^2)^{-1} \tag{7.58}$$

Signal estimation Equation:

$$\hat{x}(m) = \hat{x}(m|m-1) + K(m)\big(y(m) - \hat{x}(m|m-1)\big) \tag{7.59}$$

Covariance matrix of estimation error:

$$P(m) = [1 - K(m)]P(m|m-1) \tag{7.60}$$

Figure 7.3 shows the application of Kalman filter for estimation of a constant signal observed in noise.

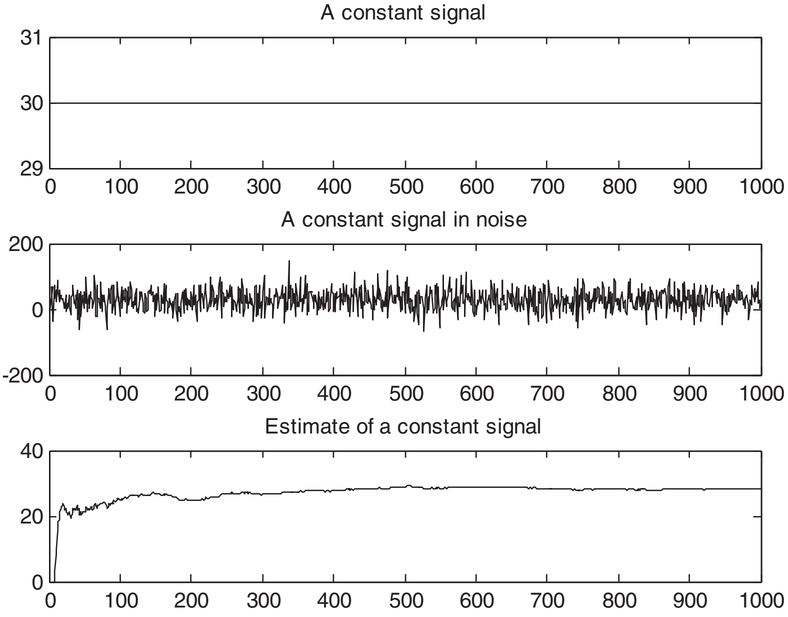


Figure 7.3 Illustration of Kalman estimate of a constant signal observed in noise.

Example 7.2 Estimation of an autoregressive (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

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + e(m) \tag{7.61}$$

$$y(m) = x(m) + n(m) \tag{7.62}$$

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. Equation (7.61) can be written in a vector form as

$$\text{Dimensions}\quad \underbrace{\begin{bmatrix} x(m) \\ x(m-1) \\ x(m-2) \\ \vdots \\ x(m-P+1) \end{bmatrix}}_{\substack{\boldsymbol{x}(m) \\ P\times 1}} = \underbrace{\begin{bmatrix} a_1 & a_2 & \cdots & a_{P-1} & a_P \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\substack{\boldsymbol{A}_x \\ P\times P}} \underbrace{\begin{bmatrix} x(m-1) \\ x(m-2) \\ x(m-3) \\ \vdots \\ x(m-P) \end{bmatrix}}_{\substack{\boldsymbol{x}(m-1) \\ P\times 1}} + \underbrace{\begin{bmatrix} e(m) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\substack{\boldsymbol{e}(m) \\ P\times 1}} \tag{7.63}$$

Assuming that $\boldsymbol{H} = \mathrm{I}$, where I is the identity matrix, and using Equation (7.63) in the Kalman filter Equations (7.5–7.11) yields the following Kalman filter algorithm:

Initial Conditions

$$\boldsymbol{P}(-1) = \delta\boldsymbol{I} \tag{7.64}$$

$$\hat{\boldsymbol{x}}(0|-1) = 0 \tag{7.65}$$

For $m = 0,\ 1,\ \ldots$

Time-Update Equations

Signal prediction Equation:

$$\hat{\boldsymbol{x}}(m|m-1) = \boldsymbol{A}\hat{\boldsymbol{x}}(m) \tag{7.66}$$

The prediction error given by

$$\begin{aligned} \boldsymbol{e}(m|m-1) &= \boldsymbol{x}(m) - \boldsymbol{A}\hat{\boldsymbol{x}}(m-1) \\ &= \boldsymbol{x}(m) - \boldsymbol{A}\big(\boldsymbol{x}(m-1) - \hat{\boldsymbol{x}}(m-1)\big) \\ &= \boldsymbol{e}(m) + \boldsymbol{A}\tilde{\boldsymbol{x}}(m-1) \end{aligned} \tag{7.67}$$

The covariance matrix of prediction error

$$\boldsymbol{P}(m|m-1) = \boldsymbol{A}\boldsymbol{P}(m-1)\boldsymbol{A}^T + \boldsymbol{Q} \tag{7.68}$$

where $\boldsymbol{Q}$ is a matrix with just one non-zero element at the top-left corner of the matrix. Taking the covariance of Equation (7.67) results in Equation (7.68).

Measurement-Update Equations

Kalman gain:

$$\boldsymbol{K}(m) = \boldsymbol{P}(m|m-1)(\boldsymbol{P}(m|m-1) + \boldsymbol{R})^{-1} \tag{7.69}$$

Signal estimation Equation:

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\big(\boldsymbol{y}(m) - \hat{\boldsymbol{x}}(m|m-1)\big) \tag{7.70}$$

Covariance matrix of estimation error:

$$\boldsymbol{P}(m) = [1 - \boldsymbol{K}(m)]\,\boldsymbol{P}(m|m-1) \tag{7.71}$$

Figure 7.4 shows an example of Kalman filtering of an AR signal observed in noise. In Chapter 17 the application of Kalman filter for restoration of speech signals in non-white noise is considered.

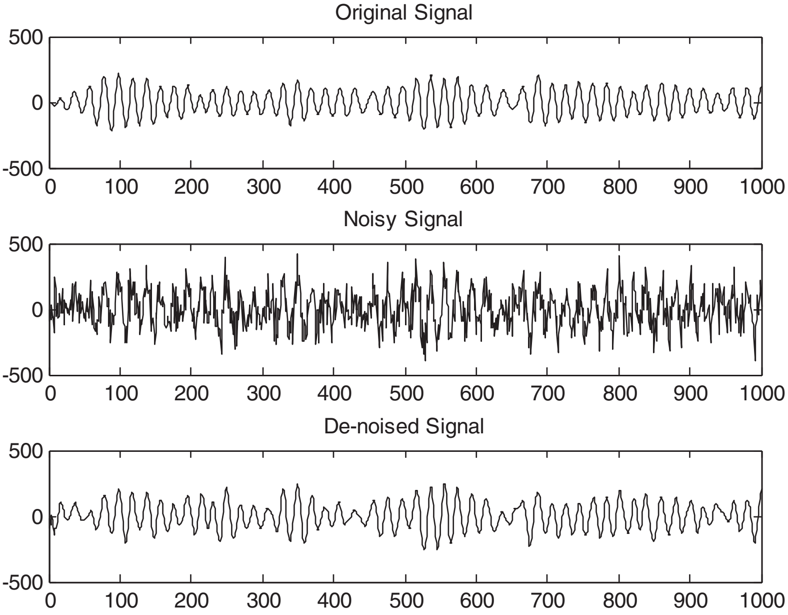


Figure 7.4 Illustration of Kalman estimation of an AR signal observed in additive white Gaussian noise.

Example 7.3 Kalman Tracker of Position and Velocity of a Vehicle

Consider a vector $\boldsymbol{x}_1(m) = [x(m), \dot{x}(m)]^T$ consisting of two elements; $\boldsymbol{x}(m)$ the position and $\dot{x}(m)$ the velocity of a moving object or vehicle. Let ΔT be the sampling period. The form of Equations for the change in position and velocity of the vehicle, in a time interval of ΔT, as given by Newton's laws of motion, are

$$\begin{gathered} position(m) = position(m-1) + \Delta T \times velocity(m-1) + \frac{\Delta T^2}{2} \times acceleration \\ velocity(m) = velocity(m-1) + \Delta T \times acceleration \end{gathered}$$

Hence the process Equation governing the evolution of the position and velocity trajectories is

$$\underbrace{\begin{bmatrix} x(m) \\ \dot{x}(m) \end{bmatrix}}_{\boldsymbol{x}_1(m)} = \underbrace{\begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}}_{\boldsymbol{F}} \underbrace{\begin{bmatrix} x(m-1) \\ \dot{x}(m-1) \end{bmatrix}}_{\boldsymbol{x}_1(m-1)} + \underbrace{\begin{bmatrix} \frac{\Delta T^2}{2} \\ \Delta T \end{bmatrix}}_{\boldsymbol{G}} a(m) \tag{7.72}$$

where the acceleration $a(m) = \ddot{x}(m)$ may be assumed to be a zero-mean Gaussian signal with a variance of σ_a^2. Equation (7.72) may be expressed in a compact form as

$$\boldsymbol{x}_1(m) = \boldsymbol{F}\boldsymbol{x}_1(m-1) + \boldsymbol{G}a(m) \tag{7.73}$$

Note that the covariance matrix of $\boldsymbol{G}a(m)$ is $\boldsymbol{Q} = \sigma_a^2 \boldsymbol{G}\boldsymbol{G}^T$. At each sampling time a noisy measurement of the position variable $\boldsymbol{x}\,(m)$ may be made for example using a GPS as

$$y(m) = \boldsymbol{H}\boldsymbol{x}_1(m) + v(m) \tag{7.74}$$

where $\boldsymbol{H} = [1\ 0]$ and it is assumed that the noise $v(m)$ is a zero-mean Gaussian random process with a variance of σ_v^2. The solution is the Kalman filter provided in Box 1.

7.3 Extended Kalman Filter (EFK)

The extended Kalman filter (EKF) is developed for cases where the state and observation functions, $f(\cdot)$ and $h(\cdot)$ are nonlinear. The general form of nonlinear state and observation Equations may be expressed as

$$\boldsymbol{x}(m) = f\big(\boldsymbol{x}(m-1)\big) + \boldsymbol{e}(m) \tag{7.75}$$

$$\boldsymbol{y}(m) = h\big(\boldsymbol{x}(m)\big) + \boldsymbol{n}(m) \tag{7.76}$$

Where $f(\cdot)$ is a known nonlinear model of the dynamics of the state variable $\boldsymbol{x}(m)$ and $h(\cdot)$ is a known nonlinear model of the observation $\boldsymbol{y}(m)$.

Using a Taylor series expansion, the nonlinear state transition function $f(\boldsymbol{x}(m-1))$ can be expressed in terms of its partial derivatives at a point $\boldsymbol{a}$. Truncating the Taylor series at the first order partial derivative term, we obtain the following approximation of Equation (7.75)

$$\boldsymbol{x}(m) \approx f(\boldsymbol{a}) + \nabla f(\boldsymbol{a})(\boldsymbol{x}(m-1) - \boldsymbol{a}) + \boldsymbol{e}(m) \tag{7.77}$$

where $\nabla f(\boldsymbol{a}) = \frac{\partial f(x)}{\partial x}|_{x=\boldsymbol{a}}$ is the Jacobian matrix, i.e. the matrix of first-order partial derivative of the outputs w.r.t. the inputs of the vector-valued function $f(\boldsymbol{x})$ evaluated at $\boldsymbol{x} = \boldsymbol{a}$.

For Equation (7.77), the function $f(\boldsymbol{a})$ is usually evaluated at the most recent estimate of the state vector $\hat{\boldsymbol{x}}(m-1)$ as:

$$\boldsymbol{x}(m) \approx f(\hat{\boldsymbol{x}}(m-1)) + \nabla f(\hat{\boldsymbol{x}}(m-1))(\boldsymbol{x}(m-1) - \hat{\boldsymbol{x}}(m-1)) + \boldsymbol{e}(m) \tag{7.78}$$

Equation (7.78) can be rearranged and rewritten in the following form:

$$\underbrace{\boldsymbol{x}(m) - f\big(\hat{\boldsymbol{x}}(m-1)\big)}_{\substack{\textit{State prediction error at } m;\\ \tilde{\boldsymbol{x}}(m|m-1)}} \approx \boldsymbol{A}\ \underbrace{\big(\boldsymbol{x}(m-1) - \hat{\boldsymbol{x}}(m-1)\big)}_{\substack{\textit{State estimation error at}\\ m-1;\ \tilde{\boldsymbol{x}}(m-1)}} + \boldsymbol{e}(m) \tag{7.79}$$

where $\boldsymbol{A}$ is the Jacobian matrix defined as

$$\boldsymbol{A} = \nabla f\big(\hat{\boldsymbol{x}}(m-1)\big) = \frac{\partial f\big(\hat{\boldsymbol{x}}(m-1)\big)}{\partial \hat{\boldsymbol{x}}(m-1)} \tag{7.80}$$

From Equation (7.79) it follows that the relationship between the covariance matrix of the prediction error vector and the estimation error vector can be expressed as

$$\boldsymbol{P}(m|m-1) = \boldsymbol{A}\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{Q} \tag{7.81}$$

Similarly, the prediction of observation can be linearised, using a Taylor series expansion of $h(f(\boldsymbol{x}(m)))$ truncated at the first term, as

$$\boldsymbol{y}(m) \approx h\Big(f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big) + \nabla h\Big(f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big)\Big(\boldsymbol{x}(m) - f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big) + \boldsymbol{n}(m) \tag{7.82}$$

Note that the Taylor series is evaluated at about the most recent prediction of the state vector $f(\hat{\boldsymbol{x}}(m-1))$. Rearranging Equation gives the observation prediction error

$$\underbrace{\boldsymbol{y}(m) - h\Big(f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big)}_{\textit{Innovation}} \approx \boldsymbol{H}\underbrace{\Big(\boldsymbol{x}(m) - f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big)}_{\textit{State prediction error}} + \boldsymbol{n}(m) \tag{7.83}$$

where $\boldsymbol{H}$ is the Jacobian matrix composed of the first order partial derivatives of the outputs w.r.t. the inputs of the vector-valued function $h(f(\hat{\boldsymbol{x}}(m-1)))$ and acts as a observation or channel distortion matrix for the error function. The Jacobian matrix $\boldsymbol{H}$ is defined as

$$\boldsymbol{H} = \nabla h\big(\hat{\boldsymbol{x}}(m-1)\big) = \frac{\partial h\big(\hat{\boldsymbol{x}}(m-1)\big)}{\partial \hat{\boldsymbol{x}}(m-1)} \tag{7.84}$$

Hence, in the extended Kalman filter the linearised transition matrix $\boldsymbol{A}$ and the linearised observation (channel) matrix $\boldsymbol{H}$ are approximated as shown in Equations (7.80) and (7.84).

The extended Kalman filter is shown in Box 2, note that the main differences with the linear Kalman filter of Box 1 are that the transition matrix $\boldsymbol{A}$ and the observation matrix $\boldsymbol{H}$ are calculated as in Equations (7.80) and (7.84), the state prediction is calculated as $\hat{\boldsymbol{x}}(m|m-1) = f\big(\hat{\boldsymbol{x}}(m-1)\big)$ and the innovation is calculated as $\boldsymbol{y}(m) - h\big(\hat{\boldsymbol{x}}(m|m-1)\big)$. Also note that in the extened Kalman filter the linearised transition and observation matrices are only used for calculation of the covariance matrices of the prediction and estimation errors and the Kalman gain. The actual prediction of the state vector and the observation vector are made using the nonlinear functions, $f(\cdot)$ and $h(\cdot)$.

Box 2 – Extended Kalman Filter Algorithm

Input: observation vectors $\{\boldsymbol{y}(m)\}$
Output: state or signal vectors $\{\hat{\boldsymbol{x}}(m)\}$

Initial Conditions
Prediction error covariance matrix:

$$\boldsymbol{P}(0|-1) = \delta\mathbf{I} \tag{7.85}$$

Prediction:

$$\hat{\boldsymbol{x}}(0|-1) = 0 \tag{7.86}$$

For $m = 0, 1, \ldots$

Time-Update Process Prediction Equations
Linearised state transition matrix model:

$$\boldsymbol{A} = \frac{\partial f\big(\hat{\boldsymbol{x}}(m-1)\big)}{\partial \hat{\boldsymbol{x}}(m-1)} \tag{7.87}$$

Box 2 – (Continued)

State prediction Equation:

$$\hat{\boldsymbol{x}}\big(m|m-1\big)=f\big(\hat{\boldsymbol{x}}(m-1)\big) \tag{7.88}$$

Covariance matrix of prediction error:

$$\boldsymbol{P}(m|m-1)=\boldsymbol{A}\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}}+\boldsymbol{Q} \tag{7.89}$$

Measurement-Update (Estimate) Equations
Linearised observation channel matrix model:

$$\boldsymbol{H}=\frac{\partial h\Big(f\big(\hat{\boldsymbol{x}}(m-1)\big)\Big)}{\partial f\big(\hat{\boldsymbol{x}}(m-1)\big)}=\frac{\partial h\big(\hat{\boldsymbol{x}}(m|m-1)\big)}{\partial\hat{\boldsymbol{x}}(m|m-1)} \tag{7.90}$$

Kalman gain vector:

$$\boldsymbol{K}(m)=\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{P}(m|m-1)\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R})^{-1} \tag{7.91}$$

State update:

$$\hat{\boldsymbol{x}}(m)=\hat{\boldsymbol{x}}(m|m-1)+\boldsymbol{K}(m)\Big(\boldsymbol{y}(m)-h\big(\hat{\boldsymbol{x}}(m|m-1)\big)\Big) \tag{7.92}$$

Covariance matrix of estimation error:

$$\boldsymbol{P}(m)=[\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H}]\boldsymbol{P}(m|m-1) \tag{7.93}$$

7.4 Unscented Kalman Filter (UFK)

The unscented Kalman filter (Julier, 1996), like the extended Kalman filter (EKF), is a variation of Kalman filter developed for nonlinear systems. The general form of nonlinear state and observation Equations may be written as

$$\boldsymbol{x}(m)=f\big(\boldsymbol{x}(m-1)\big)+\boldsymbol{e}(m) \tag{7.94}$$

$$\boldsymbol{y}(m)=h\big(\boldsymbol{x}(m)\big)+\boldsymbol{n}(m) \tag{7.95}$$

Where $f(\cdot)$ is a known nonlinear model of the dynamics of the state variable $\boldsymbol{x}(m)$ and $h(\cdot)$ is a known nonlinear model of the observation $\boldsymbol{y}(m)$.

The unscented Kalman filter (UKF) is based on the unscented transform (UT) whereby a deterministic sampling method is used to obtain a set of samples around the current estimate. After propagating the samples through the nonlinear state and observation functions, $f(\cdot)$ and $h(\cdot)$, the mean vectors and covariance matrices of the variables of interest are recalculated.

The UKF differs from the extended Kalman filter (EKF) in two main respects:

(1) Whereas in the EKF a single prediction of the state variable is propagated through the nonlinear transition and observation functions, in the UKF several predictions of the state variable, taken at a set of deterministic points around the mean, called the sigma points, are propagated through the nonlinear transition and observation functions. Consequently, in UKF, the mean and covariance values obtained from the averages of several samples are more accurate than those obtained from a single sample in EKF.

(2) In EKF the Jacobian matrix (i.e. the matrix of the first order partial derivatives of the elements of the output vector w.r.t. the elements of the input vector) of the nonlinear transition and observation

functions are used to compute the covariance error matrices and the Kalman gain vector. In contrast in UKF is a derivative-free estimation method in which the covariance error matrices and the Kalman gain vector are obtained from the samples taken around the current estimate.

The main steps for UKF method are defined next.

Step 1 – Initialisation

Define an augmented state vector $\boldsymbol{x}_a(m-1)$ and its covariance matrix $\boldsymbol{P}_a(m-1)$ as

$$\boldsymbol{x}_a(m-1) = \left[\boldsymbol{x}(m-1),\, E[\boldsymbol{e}(m)],\, E[\boldsymbol{n}(m)]\right] \tag{7.96}$$

$$\boldsymbol{P}_a(m-1) = \begin{bmatrix} \boldsymbol{P}(m-1) & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R} \end{bmatrix} \tag{7.97}$$

where the subscript ‘a’ denotes augmented, $\boldsymbol{x}(m-1)$ is the state vector, $\mathcal{E}[\boldsymbol{e}(m)] = 0$ is the mean of the process noise vector, $\mathcal{E}[\boldsymbol{n}(m)] = 0$ is the mean of the measurement noise vector, $\boldsymbol{P}(m-1)$ is the covariance matrix of estimation error, $\boldsymbol{Q}$ is the covariance matrix of process noise $\boldsymbol{e}(m)$ and $\boldsymbol{R}$ is the covariance matrix of measurement noise $\boldsymbol{n}(m)$.

Step 2 – Selecting the sigma points

These samples are selected around the current estimate $\boldsymbol{x}_a(m-1)$, at points defined by the rows of the covariance matrix $\boldsymbol{P}_a(m-1)$. For an L-dimensional augmented state vector define $2L+1$ samples, i.e. L samples on each side of the augmented state vector, $\boldsymbol{x}_a(m-1)$, as

$$\boldsymbol{\chi}_0(m-1) = \boldsymbol{x}_a(m-1) \tag{7.98}$$

$$\boldsymbol{\chi}_i(m-1) = \boldsymbol{x}_a(m-1) + \left(\sqrt{(L+\lambda)\boldsymbol{P}_a(m-1)}\right)_i \quad i = 1, \cdots, L \tag{7.99}$$

$$\boldsymbol{\chi}_i(m-1) = \boldsymbol{x}_a(m-1) - \left(\sqrt{(L+\lambda)\boldsymbol{P}_a(m-1)}\right)_{i-L} \quad i = L+1, \cdots, 2L \tag{7.100}$$

$$\lambda = \alpha^2(L+\kappa) - L \tag{7.101}$$

Where $\left(\sqrt{(L+\lambda)P_a(m-1)}\right)_i$ is obtained from the i^{th} column of the square root covariance matrix and λ is a scaling parameter. The constant α determines the spread of the sigma points around $\boldsymbol{x}_a$, it is set to a small positive value such as 0.001. The constant $\boldsymbol{\kappa}$ is also scaling parameter but is usually set to 0.

Step 3 – Transformation of sample (sigma) points through nonlinear transition function

The sigma points are propagated through the nonlinear state transition function $f(\cdot)$ to yield predictions as

$$\boldsymbol{\chi}_i(m|m-1) = f(\boldsymbol{\chi}_i(m-1)), \quad i = 0, \cdots, 2\boldsymbol{L} \tag{7.102}$$

Step 4 – Calculation of averaged predicted state vector and covariance matrix

A set of averaging weights, W_i^s and W_i^c, are defined for the state vector and covariance matrix as

$$W_0^S = \frac{\lambda}{L+\lambda} \tag{7.103}$$

$$W_0^c = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{7.104}$$

$$W_i^S = W_i^c = \frac{\lambda}{2(L+\lambda)} \tag{7.105}$$

The constant β is used to incorporate prior knowledge of the distribution of the state vector x_a (for Gaussian distributions, $\beta = 2$ is optimal). The predicted state and covariance are obtained as

$$\boldsymbol{x}_a(m|m-1) = \sum_{i=0}^{2L} W_i^S \boldsymbol{\chi}_i(m|m-1) \tag{7.106}$$

$$\boldsymbol{P}(m|m-1) = \sum_{i=0}^{2L} W_i^c [\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}_a(m|m-1)][\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}_a(m|m-1)]^T \tag{7.107}$$

Step 5 – Transformation of the sample (sigma) points through the observation function
The sigma points are propagated through the observation function as

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\Big(\boldsymbol{y}(m) - h\big(\hat{\boldsymbol{x}}(m|m-1)\big)\Big) \tag{7.108}$$

The prediction of the observation is obtained as

$$\boldsymbol{y}(m|m-1) = \sum_{i=0}^{2L} W_i^S \Upsilon_i(m|m-1) \tag{7.109}$$

Next the auto-covariance of the observation prediction error vector and the cross-covariance of the observation prediction error vector with the observation state prediction error vector are calculated as

$$\boldsymbol{P}_{yy}(m|m-1) = \sum_{i=0}^{2L} W_i^c [\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)][\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)]^T \tag{7.110}$$

$$\boldsymbol{P}_{xy}(m|m-1) = \sum_{i=0}^{2L} W_i^c [\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)][\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}(m|m-1)]^T \tag{7.111}$$

The Kalman gain is obtained as

$$\boldsymbol{K} = \frac{\boldsymbol{P}_{yy}(m|m-1)}{\boldsymbol{P}_{xy}(m|m-1)} \tag{7.112}$$

Note that the Kalman gain term in Equation (7.112) is in the form of a Wiener filter, i.e. cross covariance divided by the autocovariance. Using the Kalman gain the state estimate is obtained as:

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\Big(\boldsymbol{y}(m) - h\big(\hat{\boldsymbol{x}}(m|m-1)\big)\Big) \tag{7.113}$$

And the covariance matrix of estimation error is

$$\boldsymbol{P}(m) = \boldsymbol{P}(m|m-1) - \boldsymbol{K}\boldsymbol{P}_{yy}\boldsymbol{K}^T \tag{7.114}$$

Box 3 – Unscented Kalman Filter Algorithm

Step 1 – Initialisation

.

$$\boldsymbol{x}_a(m-1) = \big[\boldsymbol{x}(m-1), E[\boldsymbol{e}(m)], E[\boldsymbol{n}(m)]\big] \tag{7.115}$$

$$\boldsymbol{P}_a(m-1) = \begin{bmatrix} \boldsymbol{P}(m-1) & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R} \end{bmatrix} \tag{7.116}$$

Step 2 – Selecting the sigma points

$$\lambda = \alpha^2(L+\kappa) - L \tag{7.117}$$

$$\chi_0(m-1) = \boldsymbol{x}_a(m-1) \tag{7.118}$$

$$\boldsymbol{\chi}_i(m-1) = \boldsymbol{x}_a(m-1) + \left(\sqrt{(L+\lambda)\boldsymbol{P}_a(m-1)}\right)_i \quad i = 1, \cdots, L \tag{7.119}$$

$$\boldsymbol{\chi}_i(m-1) = \boldsymbol{x}_a(m-1) - \left(\sqrt{(L+\lambda)\boldsymbol{P}_a(m-1)}\right)_{i-L} \quad i = L+1, \cdots, 2L \tag{7.120}$$

Step 3 – Transformation of sigma points through transition function

.

$$\boldsymbol{\chi}_i(m|m-1) = f\big(\boldsymbol{\chi}_i(m-1)\big), \quad i = 0, \cdots, 2L \tag{7.121}$$

Step 4: Prediction of state vector and covariance matrix

.

$$W_0^S = \frac{\lambda}{L+\lambda},\; W_0^c = \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta),\; W_i^S = W_i^c = \frac{\lambda}{2(L+\lambda)} \tag{7.122}$$

$$\boldsymbol{x}_a(m|m-1) = \sum_{i=0}^{2L} W_i^S \boldsymbol{\chi}_i(m|m-1) \tag{7.123}$$

$$\boldsymbol{P}(m|m-1) = \sum_{i=0}^{2L} W_i^c[\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}_a(m|m-1)][\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}_a(m|m-1)]^T \tag{7.124}$$

Step 5: Propagation of sigma points through observation function

.

$$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\Big(\boldsymbol{y}(m) - h\big(\hat{\boldsymbol{x}}(m|m-1)\big)\Big) \tag{7.125}$$

$$\boldsymbol{y}(m|m-1) = \sum_{i=0}^{2L} W_i^S \Upsilon_i(m|m-1) \tag{7.126}$$

$$\boldsymbol{P}_{yy}(m|m-1) = \sum_{i=0}^{2L} W_i^c[\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)][\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)]^T \tag{7.127}$$

$$\boldsymbol{P}_{xy}(m|m-1) = \sum_{i=0}^{2L} W_i^c[\Upsilon_i(m|m-1) - \boldsymbol{y}(m|m-1)][\boldsymbol{\chi}_i(m|m-1) - \boldsymbol{x}(m|m-1)]^T \tag{7.128}$$

Kalman gain: $$\boldsymbol{K} = \frac{\boldsymbol{P}_{\boldsymbol{yy}}(m|m-1)}{\boldsymbol{P}_{\boldsymbol{xy}}(m|m-1)} \tag{7.129}$$

State update: $$\hat{\boldsymbol{x}}(m) = \hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\Big(\boldsymbol{y}(m) - h\big(\hat{\boldsymbol{x}}(m|m-1)\big)\Big) \tag{7.130}$$

Covariance matrix of estimation error: $$\boldsymbol{P}(m) = \boldsymbol{P}(m|m-1) - \boldsymbol{K}\boldsymbol{P}_{\boldsymbol{yy}}\boldsymbol{K}^T \tag{7.131}$$

7.5 Sample Adaptive Filters – LMS, RLS

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 non-stationary signals. These are essential characteristics in applications such as echo cancellation, adaptive delay estimation, low-delay predictive coding, noise cancellation, radar, and channel equalisation in mobile telephony, where low delay and fast tracking of time-varying processes and time-varying environments are important objectives.

Figure 7.5 illustrates the configuration of a least square error adaptive filter. At each sampling time, an adaptation algorithm adjusts the P filter coefficients $\mathbf{w}^{\mathrm{T}}(m) = [w_0(m), w_1(m), \ldots, w_{P-1}(m)]$ to minimise the difference between the filter output and 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:

$$\text{Next parameter estimate} = \text{Previous parameter estimate} + \text{Update(error)}$$

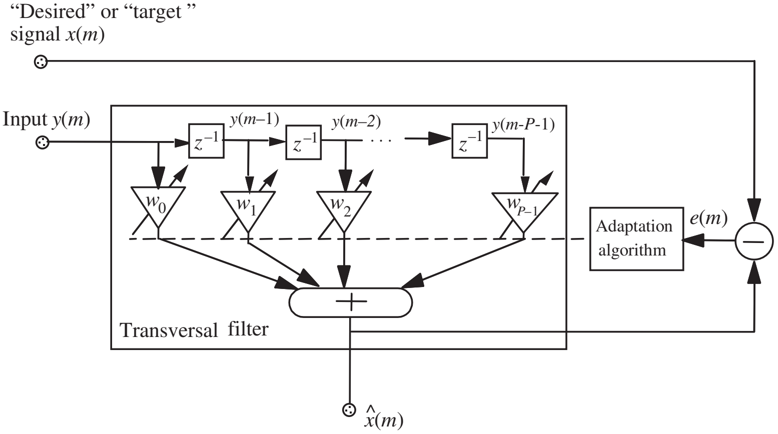


Figure 7.5 Illustration of the configuration of an adaptive filter.

where the update term is a function of the error signal. In adaptive filtering a number of decisions have to be made concerning the filter model and the adaptation algorithm:

(1) Filter type: This can be a finite impulse response (FIR) filter, or an infinite impulse response (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 it may be 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 recursive least square (RLS) error and the least mean square error (LMS) methods. The factors that influence the choice of the adaptation algorithm are the computational complexity, the speed of convergence to optimal operating conditions, the minimum error at convergence, the numerical stability and the robustness of the algorithm to initial parameter states.

(4) Optimisation criteria: In this chapter two optimality criteria are used. One is based on the minimisation of the mean of squared error (used in LMS, RLS and Kalman filter) and the other is based on constrained minimisation of the norm of the incremental change in the filter coefficients which results

in normalised LMS (NLMS). In Chapter 18, adaptive filters with non-linear objective functions are considered for independent component analysis.

7.6 Recursive Least Square (RLS) Adaptive Filters

The recursive least square error (RLS) 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 non-stationary 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 equalisation, 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 7.4 illustrates the configuration of an adaptive filter where $y(m)$, $x(m)$ and $\boldsymbol{w}^{\mathrm{T}}(m) = [w_0(m), w_1(m), \dots, w_{p-1}(m)]$ denote the filter input, the desired (target) signal and the filter coefficient vector respectively. The filter output can be expressed as

$$\hat{x}(m) = \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m) \tag{7.132}$$

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

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m) \end{aligned} \tag{7.133}$$

The adaptation process is based on the minimisation of the mean square error criterion defined as

$$\begin{aligned} \mathcal{E}[e^2(m)] &= \mathcal{E}\left\{\left[x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)\right]^2\right\} \\ &= \mathcal{E}\left[x^2(m)\right] - 2\boldsymbol{w}^{\mathrm{T}}(m)\mathcal{E}[\boldsymbol{y}(m)x(m)] + \boldsymbol{w}^{\mathrm{T}}(m)\mathcal{E}\left[\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)\right]\boldsymbol{w}(m) \\ &= r_{xx}(0) - 2\boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{r}_{\boldsymbol{yx}}(m) + \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}(m)\boldsymbol{w}(m) \end{aligned} \tag{7.134}$$

where $r_{xx}(0)$ is the autocorrelation at lag zero of the target signal $x(m)$, $\boldsymbol{R}_{yy}$ is the autocorrelation matrix of the input signal vector $\boldsymbol{y}(m)$ and $\boldsymbol{r}_{yx}$ 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 Equation (6.10), as

$$\boldsymbol{w} = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{yx} \tag{7.135}$$

In the following, we formulate a recursive, time-update, adaptive formulation of Equation (7.135). From Section 6.2, for a block of N sample vectors, the correlation matrix can be written as

$$\boldsymbol{R}_{yy} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} = \sum_{m=0}^{N-1}\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.136}$$

where $\boldsymbol{y}(m) = [y(m), \dots, y(m-P-1)]^T$. Now, the sum of vector products in Equation (7.136) can be expressed in recursive fashion as

$$\boldsymbol{R}_{yy}(m) = \boldsymbol{R}_{yy}(m-1) + \boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.137}$$

To introduce adaptability to the time variations of the signal statistics, the autocorrelation estimate in Equation (7.137) can be windowed by an exponentially decaying window:

$$\boldsymbol{R}_{yy}(m) = \lambda\boldsymbol{R}_{yy}(m-1) + \boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.138}$$

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

$$\boldsymbol{r}_{yx} = \sum_{m=0}^{N-1} \boldsymbol{y}(m)x(m) \tag{7.139}$$

The sum of products in Equation (7.137) can be calculated in recursive form as

$$\boldsymbol{r}_{yx}(m) = \boldsymbol{r}_{yx}(m-1) + \boldsymbol{y}(m)x(m) \tag{7.140}$$

Equation (7.140) can be made adaptive using an exponentially decaying forgetting factor λ:

$$\boldsymbol{r}_{yx}(m) = \lambda \boldsymbol{r}_{yx}(m-1) + \boldsymbol{y}(m)x(m) \tag{7.141}$$

For a recursive solution of the least square error Equation (7.135), we need to obtain a recursive time-update formula for the inverse matrix in the form

$$\boldsymbol{R}_{yy}^{-1}(m) = \boldsymbol{R}_{yy}^{-1}(m-1) + Update(m) \tag{7.142}$$

A recursive relation for the matrix inversion is obtained using the following lemma.

7.6.1 Matrix Inversion Lemma

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two positive-definite $P \times P$ matrices related by

$$\boldsymbol{A} = \boldsymbol{B}^{-1} + \boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^{\mathrm{T}} \tag{7.143}$$

where $\boldsymbol{D}$ is a positive-definite $N \times N$ matrix and $\boldsymbol{C}$ is a $P \times N$ matrix. The matrix inversion lemma states that the inverse of the matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A}^{-1} = \boldsymbol{B} - \boldsymbol{B}\boldsymbol{C}\left(\boldsymbol{D} + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B} \tag{7.144}$$

This lemma can be proved by multiplying Equation (7.143) and Equation (7.144). 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 $\boldsymbol{R}_{yy}^{-1}(m)$. Let

$$\boldsymbol{R}_{yy}(m) = \boldsymbol{A} \tag{7.145}$$

$$\lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1) = \boldsymbol{B} \tag{7.146}$$

$$\boldsymbol{y}(m) = \boldsymbol{C} \tag{7.147}$$

$$\boldsymbol{D} = Identity\ matrix \tag{7.148}$$

Substituting Equations (7.145) to (7.148) in Equation (7.144), we obtain

$$\boldsymbol{R}_{yy}^{-1}(m) = \lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1) - \frac{\lambda^{-2}\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)}{1 + \lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)} \tag{7.149}$$

Now define the variables $\boldsymbol{\Phi}(m)$ and $\boldsymbol{k}(m)$ as

$$\boldsymbol{\Phi}_{yy}(m) = \boldsymbol{R}_{yy}^{-1}(m) \tag{7.150}$$

and

$$\boldsymbol{k}(m) = \frac{\lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)}{1 + \lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)} \tag{7.151}$$

or

$$\boldsymbol{k}(m)=\frac{\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)}{1+\lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)} \tag{7.152}$$

Using Equations (7.152) and (7.150), the recursive Equation (7.149) for computing the inverse matrix can be written as

$$\boldsymbol{\Phi}_{yy}(m)=\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)-\lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1) \tag{7.153}$$

From Equations (7.152) and (7.153), we have

$$\begin{aligned}\boldsymbol{k}(m)&=\left[\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)-\lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\right]\boldsymbol{y}(m)\\&=\boldsymbol{\Phi}_{yy}(m)\boldsymbol{y}(m)\end{aligned} \tag{7.154}$$

Now Equations (7.153) and (7.154) are used in the following to derive the RLS adaptation algorithm.

7.6.2 Recursive Time-update of Filter Coefficients

The least square error filter coefficients are

$$\begin{aligned}\boldsymbol{w}(m)&=\boldsymbol{R}_{yy}^{-1}(m)\boldsymbol{r}_{yx}(m)\\&=\boldsymbol{\Phi}_{yy}(m)\boldsymbol{r}_{yx}(m)\end{aligned} \tag{7.155}$$

Substituting the recursive form of the correlation vector in Equation (7.155) yields

$$\begin{aligned}\boldsymbol{w}(m)&=\boldsymbol{\Phi}_{yy}(m)\left[\lambda\boldsymbol{r}_{yx}(m-1)+\boldsymbol{y}(m)x(m)\right]\\&=\lambda\boldsymbol{\Phi}_{yy}(m)\boldsymbol{r}_{yx}(m-1)+\boldsymbol{\Phi}_{yy}(m)\boldsymbol{y}(m)x(m)\end{aligned} \tag{7.156}$$

Now substitution of the recursive form of the matrix $\boldsymbol{\Phi}_{yy}(m)$ from Equation (7.153) and $\boldsymbol{k}(m)=\boldsymbol{\Phi}_{yy}(m)\boldsymbol{y}(m)$ from Equation (7.154) in the right-hand side of Equation (7.156) yields

$$\boldsymbol{w}(m)=\left[\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)-\lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\right]\lambda\boldsymbol{r}_{yx}(m-1)+\boldsymbol{k}(m)x(m) \tag{7.157}$$

or

$$\boldsymbol{w}(m)=\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{r}_{yx}(m-1)-\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{r}_{yx}(m-1)+\boldsymbol{k}(m)x(m) \tag{7.158}$$

Substitution of $\boldsymbol{w}(m-1)=\boldsymbol{\Phi}_{yy}(m-1)r_{yx}(m-1)$ in Equation (7.158) yields:

$$\boldsymbol{w}(m)=\boldsymbol{w}(m-1)+\boldsymbol{k}(m)\left[x(m)-\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{w}(m-1)\right] \tag{7.159}$$

This Equation can be rewritten in the form

$$\boldsymbol{w}(m)=\boldsymbol{w}(m-1)+\boldsymbol{k}(m)e(m) \tag{7.160}$$

Equation (7.161) is a recursive time-update implementation of the least square error Wiener filter.

Figure 7.6 shows the application of RLS filter to channel identification given the input signal and the noisy channel output signal.

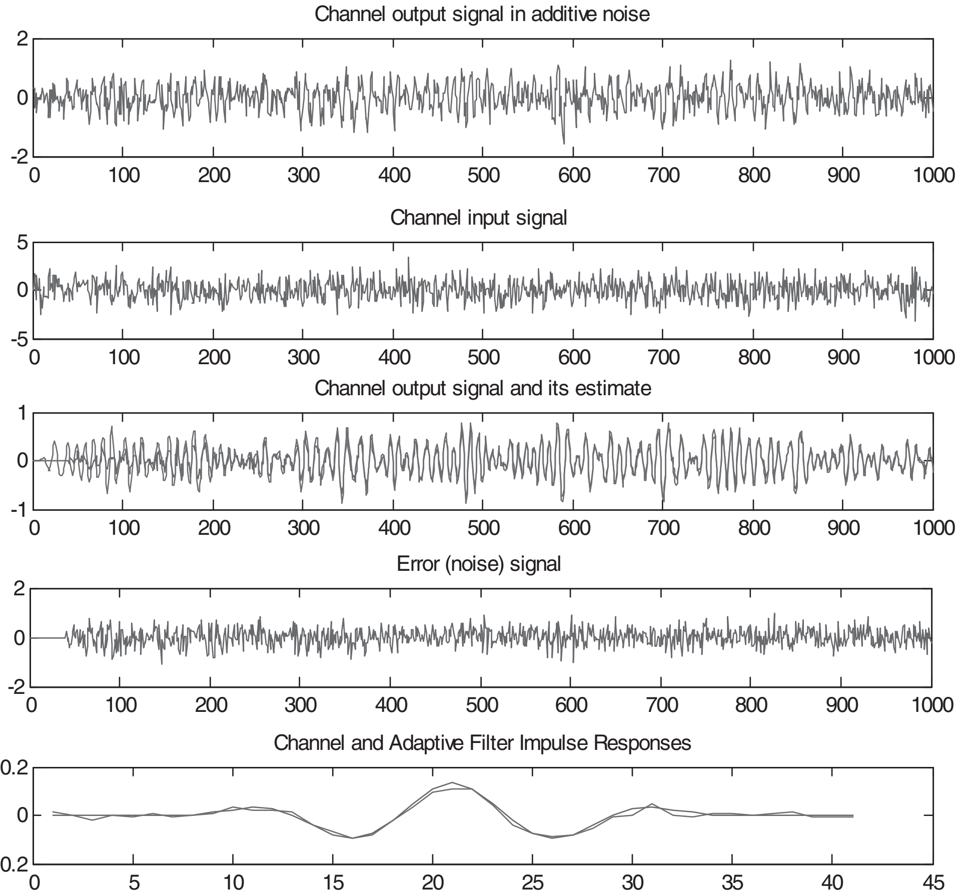


Figure 7.6 From top panel, reference signal (channel output observed in additive white Gaussian noise), input to channel, channel output and its estimate superimposed, error (and noise) signal, the channel response and its estimate.

Box 4 – RLS Adaptation Algorithm

Input signals: $y(m)$ and $x(m)$

Initial values:

$$\boldsymbol{\Phi}_{yy}\ (m) = \delta\mathbf{I}$$

$$\boldsymbol{w}(0) = \boldsymbol{w}_{\mathrm{I}}$$

For $m = 1, 2, \ldots$

Filter gain vector update:

$$\boldsymbol{k}(m) = \frac{\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)}{1+\lambda^{-1}\mathbf{y}^{\mathrm{T}}(m)\Phi_{yy}(m-1)\boldsymbol{y}(m)} \tag{7.161}$$

Error signal Equation:

$$e(m) = x(m) - \boldsymbol{w}^{\mathrm{T}}(m-1)\boldsymbol{y}(m) \tag{7.162}$$

Filter coefficients adaptation:

$$\boldsymbol{w}(m) = \boldsymbol{w}(m-1) + \boldsymbol{k}(m)e(m) \tag{7.163}$$

Inverse correlation matrix update:

$$\boldsymbol{\Phi}_{yy}(m) = \lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1) - \lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^T(m)\boldsymbol{\Phi}_{yy}(m-1) \tag{7.164}$$

7.7 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 7.7 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 the 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

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\left[-\frac{\partial \mathcal{E}[e^2(m)]}{\partial \boldsymbol{w}(m)}\right] \tag{7.165}$$

where μ is the adaptation step size, it controls the size of the update increment at each iteration or time sample.

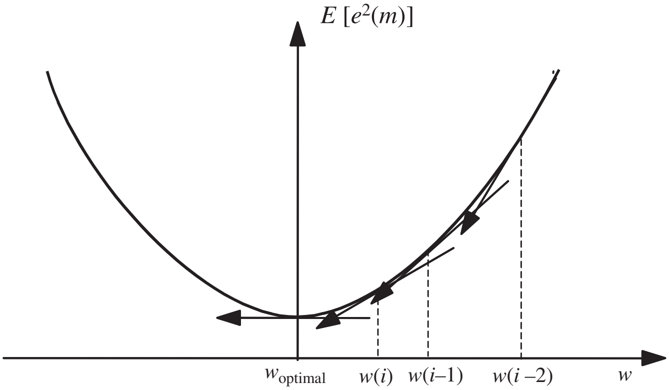


Figure 7.7 Illustration of gradient search of the mean square error surface for the minimum error point.

From Equation (7.134), the gradient (derivative) of the mean square error function is given by

$$\frac{\partial \mathcal{E}[e^2(m)]}{\partial \boldsymbol{w}(m)} = -2\boldsymbol{r}_{yx} + 2\boldsymbol{R}_{yy}\boldsymbol{w}(m) \tag{7.166}$$

Substituting Equation (7.166) in Equation (7.165) yields

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\left[\boldsymbol{r}_{yx} - R_{yy}\boldsymbol{w}(m)\right] \tag{7.167}$$

where the factor of 2 in Equation (7.166) has been absorbed in the adaptation step size μ. Let $\boldsymbol{w}_o$ denote the optimal LSE filter coefficient vector; we define a filter coefficients error vector $\tilde{\boldsymbol{w}}(m)$ as

$$\tilde{\boldsymbol{w}}(m) = \boldsymbol{w}(m) - \boldsymbol{w}_o \tag{7.168}$$

For a stationary process, the optimal LSE filter *wo* is obtained from the Wiener filter, Equation (6.10), as

$$\boldsymbol{w}_o = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{yx} \tag{7.169}$$

Note from a comparison of Equations (7.167) and (7.169) that the recursive solution (7.167) does not need the computation of the inverse of the autocorrelation matrix needed in the closed-form solution (7.169).

Subtracting $\boldsymbol{w}_o$ from both sides of Equation (7.167), and then substituting $\boldsymbol{R}_{yy}\boldsymbol{w_o}$ for $\boldsymbol{r}_{yx}$, and using Equation (7.169) yields

$$\tilde{\boldsymbol{w}}(m+1) = \left[\mathbf{I} - \mu\boldsymbol{R}_{yy}\right]\tilde{\boldsymbol{w}}(m) \tag{7.170}$$

It is desirable that the filter error vector $\tilde{\boldsymbol{w}}(m)$ 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 (7.170), 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

$$\boldsymbol{R}_{yy} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}} \tag{7.171}$$

where $\boldsymbol{Q}$ is an orthonormal matrix of the eigenvectors of $\boldsymbol{R}_{yy}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix with its diagonal elements corresponding to the eigenvalues of $\boldsymbol{R}_{yy}$. Substituting $\boldsymbol{R}_{yy}$ from Equation (7.171) in Equation (7.170) yields

$$\tilde{\boldsymbol{w}}(m+1) = \left[\mathbf{I} - \mu\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}\right]\tilde{\boldsymbol{w}}(m) \tag{7.172}$$

Multiplying both sides of Equation (7.172) by *QT* and using the relation $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}} = \mathbf{I}$ yields

$$\boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m+1) = [\mathbf{I} - \mu\boldsymbol{\Lambda}]\boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m) \tag{7.173}$$

Let

$$\boldsymbol{v}(m) = \boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m) \tag{7.174}$$

Then

$$\boldsymbol{v}(m+1) = [\mathbf{I} - \mu\boldsymbol{\Lambda}]\,\boldsymbol{v}(m) \tag{7.175}$$

As the matrices $\boldsymbol{\Lambda}$ and $\mathbf{I}$ are both diagonal, Equation (7.175) can be expressed in terms of the Equations for the individual elements of the error vector $\boldsymbol{v}(m)$ as

$$v_k(m+1) = [1 - \mu\lambda_k]\,v_k(m) \tag{7.176}$$

where λ_k is the k$^{\mathrm{th}}$ eigenvalue of the autocorrelation matrix of the filter input y(m). Figure 7.8 is a feedback network model of the time variations of the error vector. From Equation (7.176), the condition for the stability of the adaptation process and the decay of the coefficient error vector is

$$-1 < 1 - \mu\lambda_k < 1 \tag{7.177}$$

Let $\lambda_{\max}$ denote the maximum eigenvalue of the autocorrelation matrix of $y(m)$, then from Equation (7.177) the limits on μ for stable adaptation are given by

$$0 < \mu < \frac{2}{\lambda_{\max}} \tag{7.178}$$

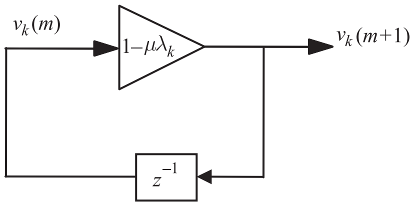


Figure 7.8 A feedback model of the variation of coefficient error with time.

7.7.1 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 errors in filter coefficients with maximum and minimum eigenvalues, $\lambda_{\max}$ and $\lambda_{\min}$ converge according to the Equations:

$$v_{\max}(m+1) = (1-\mu\lambda_{\max})\, v_{\max}(m) \tag{7.179}$$

$$v_{\min}(m+1) = (1-\mu\lambda_{\min})\, v_{\min}(m) \tag{7.180}$$

The ratio of the maximum to the minimum eigenvalue of a correlation matrix is called the eigenvalue spread of the correlation matrix:

$$\text{eigenvaluespread} = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{7.181}$$

Note that the differences in the speed of convergence of filter coefficients is proportional to the spread in eigenvalue of the autocorrelation matrix of the input signal. Also note that as the adaptation step size μ affects the convergence rate and as it must be less that $2/\lambda_{\max}$, if $\lambda_{\max}$ is reduced μ can be increased resulting in a more even and faster convergence rate. This can be achieved by pre-whitening the signal using an inverse linear prediction filter or by the application of the adaptive filters in sub-bands where the signal within each sub-band has a smaller eigenvalue spread.

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.

Solution:

(1) Eigenvalue spread $= \dfrac{\lambda_{\max}}{\lambda_{\min}} = 10$
(2) The bounds on adaptation step size. $0<\mu<\dfrac{2}{\lambda_{\max}}=1$
(3) The fastest decay factor $= (1-\mu\ \lambda_{\max}) = 1-0.4\times 2 = 0.2$, and the slowest decay factor $= (1-\mu\ \lambda_{\min}) = 1-0.4\times 0.2 = 0.92$.

7.7.2 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 $\boldsymbol{\mu} = [\mu_0, \ \mu_2, \ \ldots, \ \mu_{P-1}]^{\mathrm{T}}$ 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.

7.8 Least Mean Squared Error (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 least mean square (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

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\left(-\frac{\partial e^2(m)}{\partial \boldsymbol{w}(m)}\right) \tag{7.182}$$

where the error signal $e(m)$ is the difference between the adaptive filter output and the target (desired) signal $\boldsymbol{x}(m)$, given by

$$e(m) = x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m) \tag{7.183}$$

The instantaneous gradient of the squared error can be re-expressed as

$$\begin{aligned}\frac{\partial e^2(m)}{\partial \boldsymbol{w}(m)} &= \frac{\partial}{\partial \boldsymbol{w}(m)}[x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)]^2 \\ &= -2\boldsymbol{y}(m)[x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)] \\ &= -2\boldsymbol{y}(m)e(m)\end{aligned} \tag{7.184}$$

Substituting Equation (7.184) into the recursion filter update Equation (7.182) and absorbing the factor of 2 in the adaptation step size μ yields the LMS adaptation Equation

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\,[\boldsymbol{y}(m)e(m)] \tag{7.185}$$

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.

7.8.1 Leaky LMS Algorithm

The stability and the adaptability of the recursive LMS adaptation Equation (7.185) can be improved by introducing a so-called leakage factor α as

$$\boldsymbol{w}(m+1) = \alpha\boldsymbol{w}(m) + \mu\,[\boldsymbol{y}(m)e(m)] \tag{7.186}$$

Note that, as illustrated in Figure 7.9, the feedback Equation for the time update of the filter coefficients is essentially a recursive (infinite impulse response) system with input $\mu\boldsymbol{y}(m)e(m)$ and its poles at α.

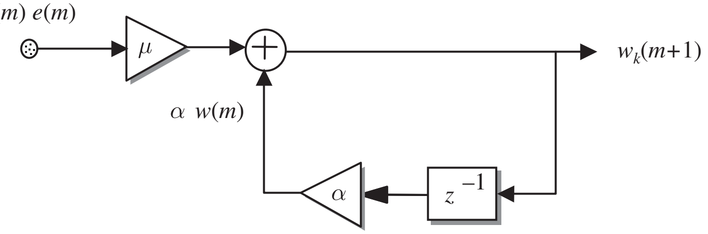


Figure 7.9 Illustration of leaky LMS adaptation of a filter coefficient.

When the parameter $\alpha < 1$, the effect is to introduce more stability and accelerate the filter adaptation to the changes in input signal characteristics.

7.8.2 Normalised LMS Algorithm

The normalised LMS (NLMS) adaptation Equation is given by

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \frac{\mu}{a + \sum_{k=0}^{P-1} y^2(m-k)} \boldsymbol{y}(m)e(m) \tag{7.187}$$

where $\sum_{k=0}^{P-1} y^2(m-k)$ is the input signal energy, μ controls the adaptation step size and a is a small constant employed to avoid the denominator of the update term becoming zero when the input signal $y(m)$ is zero.

7.8.2.1 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 $\delta\boldsymbol{w}(m+1)$ in the successive updates of the filter coefficient vector is used:

$$\|\delta\boldsymbol{w}(m+1)\| = \sum_{k=0}^{P-1} \left(w_k(m+1) - w_k(m)\right)^2 \tag{7.188}$$

subject to the constraint that

$$\boldsymbol{w}^{\mathrm{T}}(m+1)\,\boldsymbol{y}(m) = x(m) \tag{7.189}$$

The solution that satisfies the above criterion and the constraint can be obtained by the Lagrange multipliers method.

In the application of the Lagrange method to the above problem, we define an optimisation criterion $J(\cdot)$ as a combination of the squared error criterion expressed in Equation (7.188) and the constraint expressed in Equation (7.189) as

$$\begin{aligned} J(\boldsymbol{w}(m+1),\lambda) &= \|\delta\boldsymbol{w}(m+1)\| + \lambda[x(m) - \boldsymbol{w}^{\mathrm{T}}(m+1)\boldsymbol{y}(m)] \\ &= \sum_{k=0}^{P-1}[w_k(m+1) - w_k(m)]^2 + \lambda[x(m) - \sum_{k=0}^{P-1} w_k(m+1)y(m-k)] \end{aligned} \tag{7.190}$$

where λ is a Lagrange multiplier. To obtain the minimum of the criterion $J(\boldsymbol{w}(m+1),\lambda)$ set the derivative of J with respect to each coefficient $\mathrm{w_i(m+1)}$ to zero as

$$\frac{\partial J(\boldsymbol{w}(m+1),\lambda)}{\partial w_i(m+1)} = 2w_i(m+1) - 2w_i(m) - \lambda y(m-i) = 0 \quad i=0,\ldots,P-1 \tag{7.191}$$

From Equation (7.191) we have

$$w_i(m+1) = w_i(m) + \frac{\lambda}{2} y(m-i) \tag{7.192}$$

From Equations (7.192) and (7.189) we have

$$\sum_{m=0}^{P-1} w_i(m) y(m-i) + \frac{\lambda}{2} \sum_{m=0}^{P-1} y(m-i)^2 = x(m) \tag{7.193}$$

Hence the Lagrange multiplier parameter λ is given by

$$\lambda = \frac{2[x(m) - \sum\limits_{k=0}^{P-1} w_k(m) y(m-k)]}{\sum\limits_{k=0}^{P-1} y^2(m-k)} = \frac{2e(m)}{\sum\limits_{k=0}^{P-1} y^2(m-k)} \tag{7.194}$$

Substitution for λ from Equation (7.194) in Equation (7.192) yields the NLMS Equation

$$w_i(m+1) = w_i(m) + \frac{e(m)}{\sum\limits_{k=0}^{P-1} y^2(m-k)} y(m-i) \tag{7.195}$$

The NLMS Equation (7.187) is obtained from (7.195) by introducing a variable μ to control the step size and a variable a to prevent the denominator of the update term Equation (7.195) becoming zero when the input signal vector $\boldsymbol{y}$ has zero values.

7.8.2.2 Steady-State Error in LMS

The least mean square error (LSE), $E_{\min}$, is achieved when the filter coefficients approach the optimum value defined by the block least square error Equation $\boldsymbol{w}_{\mathrm{o}} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{yx}$ derived in Chapter 6. The steepest-decent 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 non-zero such as noise reduction, the incremental update term $\mu \boldsymbol{y}(m)e(m)$ would remain non-zero even when the optimal point is 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.

Figure 7.10 Illustrates the application of NLMS filter to adaptive noise canellation where the reference and input both contain the desired signal with different levels of SNRs.

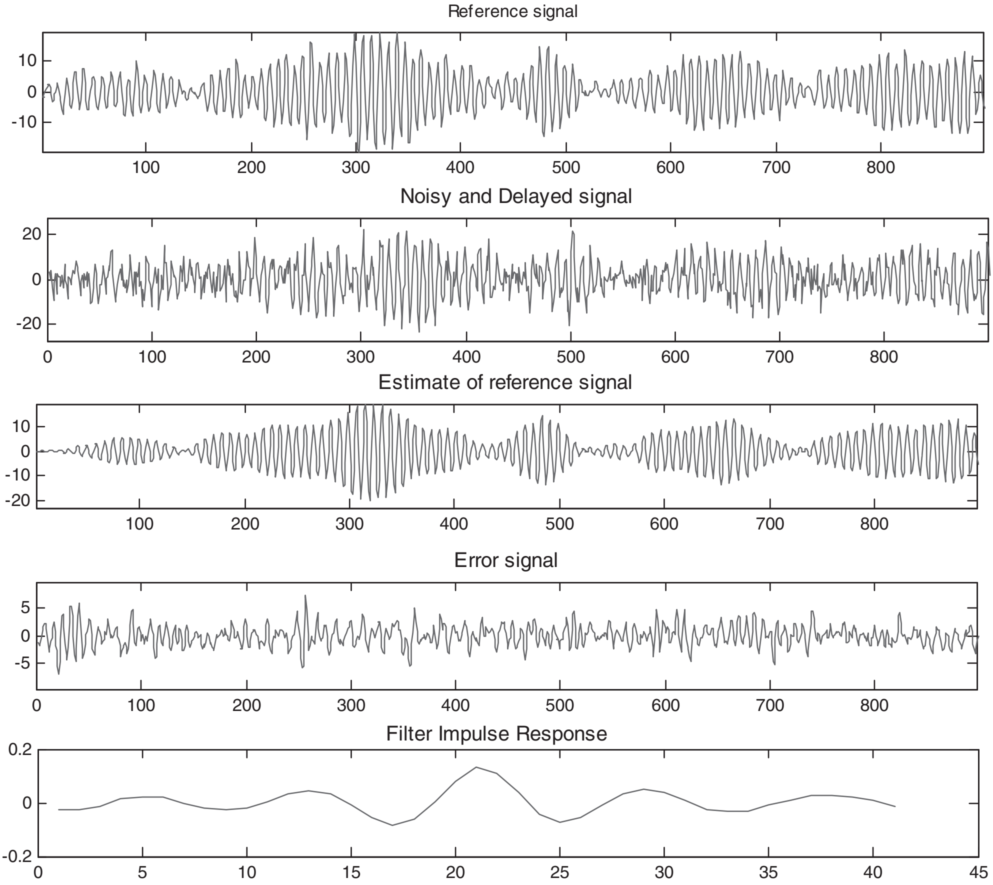


Figure 7.10 Illustration of the inputs and outputs of NLMS adaptation. The filter input and the target/reference signal both contain the clean signal with different SNRs. The delay between target and reference signal is 20 samples as evident from the adaptive filter's impulse response.

7.9 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 least squared error criterion, the Kalman filter differs from the Wiener filter in two respects. First, the Kalman filter can be applied to non-stationary 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, time-delay estimation, etc., the RLS and LMS filters provide a suitable alternative to the Kalman filter. The RLS filter is a recursive implementation of the solution for 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 non-stationary with relatively high rate of change (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 less sensitivity to the eigenvalue spread, becomes a more attractive alternative.

Bibliography

Alexander S.T. (1986) *Adaptive Signal Processing: Theory and Applications*. Springer-Verlag, 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. Acoustics Speech and Signal Processing*, **ASSP-34**: 793–809.

Bershad N.J. and QU L.Z. (1989) On the Probability Density Function of the LMS Adaptive Filter Weights. *IEEE Trans. Acoustics Speech and Signal Processing*, **ASSP–37**: 43–59.

Bozic, S.M., *Digital and Kalman Filtering*. Edward Arnald Publications, 1979 (1st edn), 1994 (2nd edn).

Bucy, R.S. and Joseph, P.D, *Filtering for Stochastic Processes with Applications to Guidance*, John Wiley & Sons, Inc, 1968; 2nd edn, AMS Chelsea Publ., 2005.

Cioffi J.M. and Kailath T. (1984) Fast Recursive Least Squares transversal filters for adaptive filtering. *IEEE Trans. Acoustics Speech and Signal Processing*, **ASSP-32**, April: 304–339.

Classen T.A. and Mecklanbrauker W.F. (1985) Adaptive Techniques for Signal Processing in Communications. *IEEE Communications*, **23**: 8–19.

Cowan C.F. and Grant P.M. (1985) *Adaptive Filters*. Prentice-Hall, Englewood Cliffs, NJ.

Duncan, D.B. and Horn S.D. (1972) Linear Dynamic Recursive Estimation from the Viewpoint of Regression Analysis. *Journal of the American Statistical Association*, **67**: 815–821.

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 Non-linear 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, Hingham, MA.

Julier, S.J., Uhlmann, J.K. (1996). A General Method for Approximating Nonlinear Transformations of Probability Distributions (Technical Report). Oxford, UK: Robotics Research Group, Department of Engineering Science, University of Oxford.

T. Kailath, (1981) *Lecture Notes on Wiener and Kalman Filtering*. Springer-Verlag.

Kailath T., Sayed, A.H., & Hassibi, B. (2000). *Linear Estimation*. Upper Saddle River, NJ USA: Prentice Hall.

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. of the ASME, Series D, *Journal of Basic Engineering*, **82**: 34–45.

Kalman R.E. and Bucy R.S. (1961) New Results in Linear Filtering and Prediction Theory. Trans. *ASME J. Basic Eng*., **83**: 95–108.

Maybeck P.S. (1990) *The Kalman Filter: An Introduction to Concepts in Autonomous Robot Vehicles*, I.J. Cox, G.T. Wilfong (eds), Springer-Verlag,.

Lauritzen, *S.L. Thiele: Pioneer in Statistics*. Oxford University Press, 2002.

Ribeiro M.I. (2004) Kalman and Extended Kalman Filters: Concept, Derivation and Properties, Institute for Systems and Robotics, Instituto Lisbon, Portugal, http://users.isr.ist.utl.pt/∼mir/.

Stratonovich, R.L. (1959). Optimum nonlinear systems which bring about a separation of a signal with constant parameters from noise. *Radiofizika*, **2**(6) 892–901.

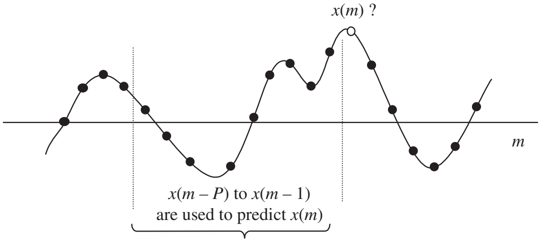
Stratonovich, R.L. (1960) Application of the Markov processes theory to optimal filtering. *Radio Engineering and Electronic Physics*, **5**(11): 1–19.

Widrow B. (1990) 30 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

Linear prediction (LP) models predicts/forecast the future values of a signal from a linear combination of their past values. A linear predictor model is an all-pole filter that models the resonances (poles) of the spectral envelope of a signal or a system.

LP models are used in diverse areas of 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 utilises 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 are considered.

8.1 Linear Prediction Coding

As illustrated in Figures 8.1 and 8.2, a linear prediction model is an all-pole filter that forecasts the future values of a signal from a linear combination of its past values.

In terms of usefulness and the range of applications as a signal processing tool, the linear prediction (LP) model ranks alongside the Fourier transform. A major large-volume application of the LP model is in digital mobile phones where voice coders use linear prediction models for the efficient coding of speech. LP models are also used in inter-frame coding of music and image coding.

There are two main motivations for the use of predictors in signal processing applications:

(1) To predict the trajectory of a signal. In the frequency domain the trajectory prediction is equivalent to the modelling of the spectrum of the signal.
(2) To remove the predictable part of a signal in order to avoid retransmitting ‘redundant’ parts of a signal that can be predicted at the receiver and thereby save storage, bandwidth, time and power.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

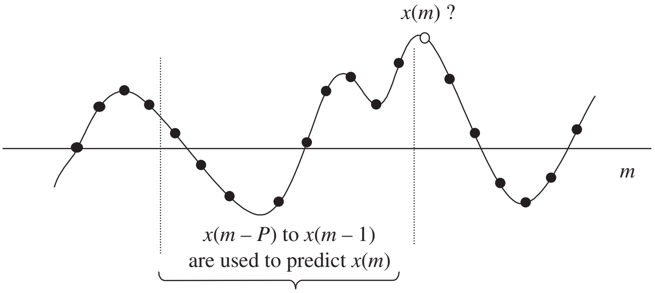


Figure 8.1 Illustration of prediction of a sample $x(m)$ from P past samples.

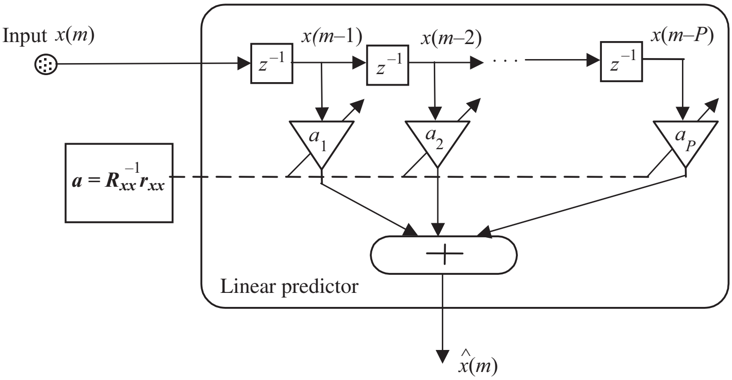


Figure 8.2 Illustration of a linear predictor: $x(m)$ is predicted as the product of the input vector $[x(m-1), \ldots, x(m-P)]$ and the predictor coefficients $[a_1, \ldots, a_P]$. The predictor coefficients are obtained from $\boldsymbol{a} = \boldsymbol{R}_{xx}^{-1}\boldsymbol{r}_{xx}$ as derived in this chapter.

8.1.1 Predictability, Information and Bandwidth

The accuracy by 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 8.3,

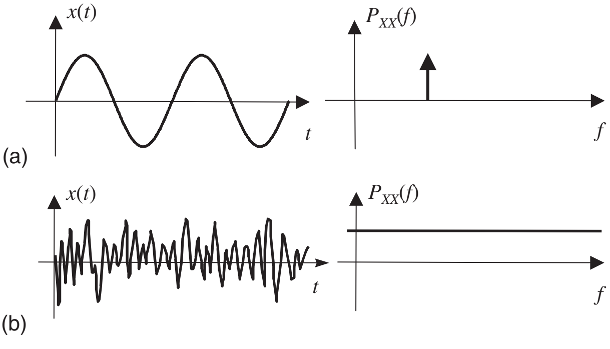


Figure 8.3 The concentration or spread of power in frequency indicates the predictable or random character of a signal: (a) a predictable signal; (b) a random signal.

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 extremely 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.

8.1.2 Applications of LP Model in Speech Processing

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 of the vocal tract. Figure 8.4 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 quasi-periodic excitation pulses for voiced sounds as shown in Figure 8.4. 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 8 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.

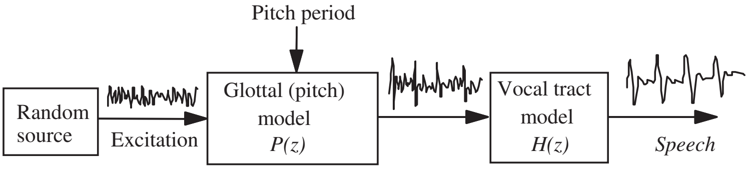


Figure 8.4 A source-filter model of speech production. The filter is usually modelled with a linear prediction model.

8.1.3 Time-Domain Description of LP Models

A linear predictor model, Figures 8.1, 8.2 and 8.5, forecasts the amplitude of a signal at time m, $x(m)$, using a linearly weighted combination of P past samples $[x(m-1),\ x(m-2),\ \ldots,\ x(m-P)]$ 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 (8.1) is illustrated in Figure 8.4.

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 feedback Equation

$$x(m) = \underbrace{\sum_{k=1}^{P} a_k x(m-k)}_{\text{Predictable part of } x(m)} + \underbrace{e(m)}_{\substack{\text{Unpredictable} \\ \text{part of } x(m)}} \tag{8.3}$$

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, unit-variance random signal, and G, a gain term, is the square root of the variance (i.e. power) of $e(m)$:

$$G = \left(\mathcal{E}\,[e^2(m)]\right)^{1/2} \tag{8.4}$$

where $E[\cdot]$ is an averaging, or expectation, operator.

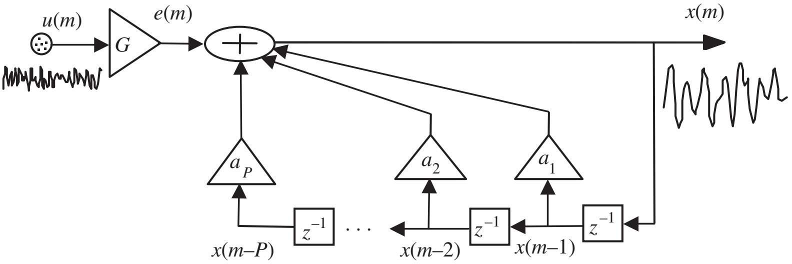


Figure 8.5 Illustration of a signal generated by a linear predictive model.

8.1.4 Frequency Response of LP Model and its Poles

The z-transform of LP Equation (8.3) shows that the LP model is an all-pole digital filter with transfer function

$$H(z) = \frac{X(z)}{U(z)} = \underbrace{\frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}}}_{\text{Cartesian form}} = \underbrace{G \frac{1}{\prod_{k=1}^{N}(1 - r_k z^{-1})} \frac{1}{\prod_{k=1}^{M}(1 - 2r_k \cos\varphi_k z^{-1} + r_k^2 z^{-2})}}_{\text{Polar form}} \tag{8.5}$$

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 are the radius and angle of the k^{th} pole. The frequency response of an LP model is given by

$$H(f) = \frac{G}{1 - \sum_{k=1}^{P} a_k e^{-j2\pi kf}} = G \frac{1}{\prod_{k=1}^{N} (1 - r_k e^{-j2\pi f})} \frac{1}{\prod_{k=1}^{M} (1 - 2r_k \cos\varphi_k e^{-j2\pi f} + r_k^2 e^{-j4\pi f})} \tag{8.6}$$

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

$$z_k = r_k e^{\pm j\varphi_k} \tag{8.7}$$

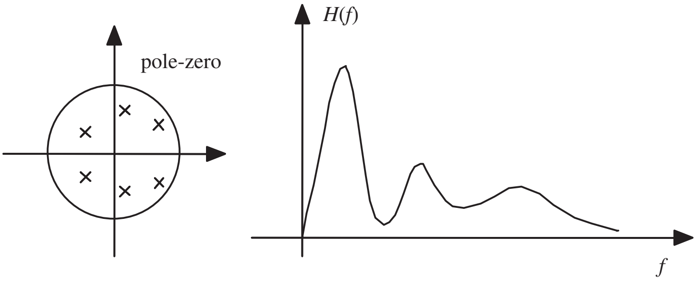


Figure 8.6 The pole–zero position and frequency response of a linear predictor.

The resonance frequency of a complex pair of poles is given by

$$F(\varphi_k) = \frac{F_s}{2\pi} \varphi_k \tag{8.8}$$

where F_s is the sampling frequency. The bandwidth of a pole is related to its radius r_k as

$$B_k = (-\log r_k)(F_s/\pi) \tag{8.9}$$

Note that the radius of a pole affects the bandwidth and damping of the resonance; these parameters are related to the amount of resistance to oscillations at the pole frequency. As the pole radius decreases the bandwidth and damping increase. When the radius of a pair of complex conjugate poles is equal to one, then the resistance to oscillations is zero, and the bandwidth of the resonance is zero, the damping is zero and the impulse response of the pole is a sine wave. The magnitude of the spectral resonance at the pole is given by $H(f = \varphi)$.

8.1.5 Calculation of Linear Predictor Coefficients

Linear predictor coefficients are obtained by minimising the mean square prediction error as

$$\begin{aligned}
\mathcal{E}[e^2(m)] &= \mathcal{E}\left[\left(x(m)-\sum_{k=1}^{P} a_k x(m-k)\right)^2\right]\\
&= \mathcal{E}\left[x^2(m)\right]-2\sum_{k=1}^{P} a_k\ \mathcal{E}[x(m)x(m-k)]+\sum_{k=1}^{P} a_k \sum_{j=1}^{P} a_j\ \mathcal{E}[x(m-k)x(m-j)]\\
&= r_{xx}(0)-2\sum_{k=1}^{P} a_k\ r_{xx}(k)+\sum_{k=1}^{P} a_k \sum_{j=1}^{P} a_j\ r_{xx}(k-j)\\
&= r_{xx}(0)-2\boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{T}}\boldsymbol{a}+\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{a}
\end{aligned} \tag{8.10}$$

where $\boldsymbol{R}_{\boldsymbol{xx}} = E[\boldsymbol{xx}^{\mathrm{T}}]$ is the autocorrelation matrix of the input vector $\boldsymbol{x}^{\mathrm{T}} = [x(m-1),\ x(m-2),\ldots,\ x(m-P)]$, $\boldsymbol{r}_{\boldsymbol{xx}} = E[x(m)\boldsymbol{x}]$ is the autocorrelation vector and $\boldsymbol{a}^{\mathrm{T}} = [a_1,\ a_2,\ldots,\ a_P]$ is the predictor coefficient vector. From Equation (8.10), the gradient of the mean square prediction error with respect to the predictor coefficient vector $\boldsymbol{a}$ is given by

$$\frac{\partial}{\partial \boldsymbol{a}}\mathcal{E}[e^2(m)] = -2\boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{T}}+2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}_{\boldsymbol{xx}} \tag{8.11}$$

where the gradient vector is defined as

$$\frac{\partial}{\partial \boldsymbol{a}} = \left(\frac{\partial}{\partial\, a_1},\frac{\partial}{\partial\, a_2},\ldots,\frac{\partial}{\partial\, a_P}\right)^{\mathrm{T}} \tag{8.12}$$

The least mean square error solution, obtained by setting Equation (8.11) to zero, is given by

$$\boldsymbol{R}_{\boldsymbol{xx}}\,\boldsymbol{a} = \boldsymbol{r}_{\boldsymbol{xx}} \tag{8.13}$$

From Equation (8.13) the predictor coefficient vector is given by

$$\boldsymbol{a} = \boldsymbol{R}_{\boldsymbol{xx}}^{-1}\boldsymbol{r}_{\boldsymbol{xx}} \tag{8.14}$$

Equation (8.14) may also be written in an expanded form as

$$\begin{pmatrix} a_1\\ a_2\\ a_3\\ \vdots\\ a_P \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1)\\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2)\\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3)\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{xx}(1)\\ r_{xx}(2)\\ r_{xx}(3)\\ \vdots\\ r_{xx}(P) \end{pmatrix} \tag{8.15}$$

An alternative formulation of the least square error problem is as follows. For a signal segment of N samples $[x(0),\ldots,\ x(N-1)]$, 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 $\boldsymbol{x}^{\mathrm{T}} = [x(-1), \ \ldots, \ x(-P)]$ is the initial vector. In compact vector/matrix notation Equation (8.16) can be written as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{Xa} \tag{8.17}$$

Using Equation (8.17), the sum of squared prediction errors over a block of N samples can be expressed as

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Xa} + \boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Xa} \tag{8.18}$$

The least squared error predictor is obtained by setting the derivative of Equation (8.14) with respect to the parameter vector $\boldsymbol{a}$ to zero:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{a}} = -2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X} + 2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = 0 \tag{8.19}$$

From Equation (8.19), the least square error predictor is given by

$$\boldsymbol{a} = \left(\boldsymbol{X}^{T}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{T}\boldsymbol{x}\right) \tag{8.20}$$

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 $\boldsymbol{R_{xx}}$. 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 $[r_{xx}(0), \ r_{xx}(1), \ldots, \ r_{xx}(P)]$ 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.

8.1.6 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.

8.1.7 The Inverse Filter: Spectral Whitening, De-correlation

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

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=1}^{P} a_k \, x(m-k) \\ &= (\boldsymbol{a}^{\text{inv}})^{\text{T}} \boldsymbol{x} \end{aligned} \tag{8.23}$$

where the inverse filter $(\boldsymbol{a}^{\text{inv}})^{\text{T}} = [1, \ -a_1, \ \ldots, \ -a_P] = [1, \ -\boldsymbol{a}]$, and $\boldsymbol{x}^{\text{T}} = [x(m), \ldots, \ x(m-P)]$. The z-transfer function of the inverse predictor model is given by

$$A(z) = 1 - \sum_{k=1}^{P} a_k \, z^{-k} \tag{8.24}$$

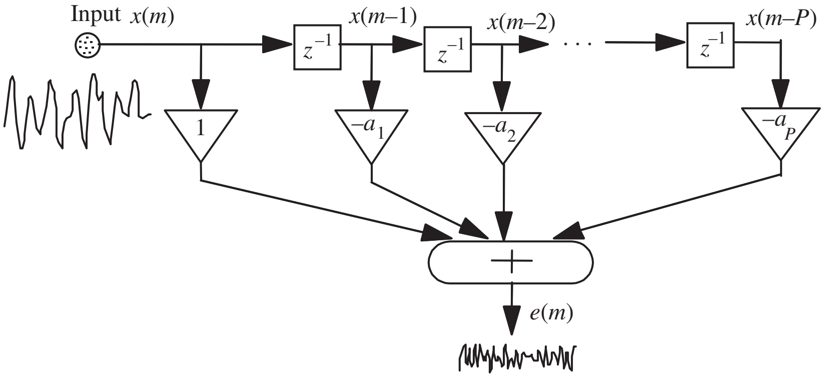


Figure 8.7 Illustration of the inverse (or whitening) filter.

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.

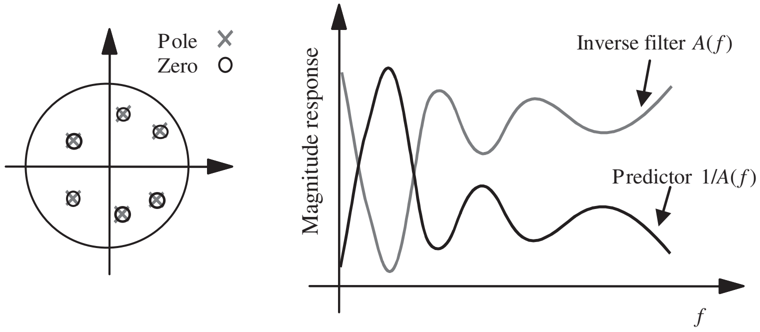


Figure 8.8 Illustration of the pole-zero diagram, and the frequency responses of an all-pole predictor and its all-zero inverse filter.

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.

8.1.8 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: (a) the signal is deterministic, (b) the signal is correctly modelled by a predictor of order P, and (c) 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 non-zero 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

$$E^{(P)} = \mathcal{E}[e^2(m)] = r_{xx}(0) - \sum_{k=1}^{P} a_k\, r_{xx}(k) \tag{8.25}$$

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

$$\mathcal{E}[e(m)e(m-k)] = \begin{cases} \sigma_e^2 = G^2 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases} \tag{8.26}$$

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.

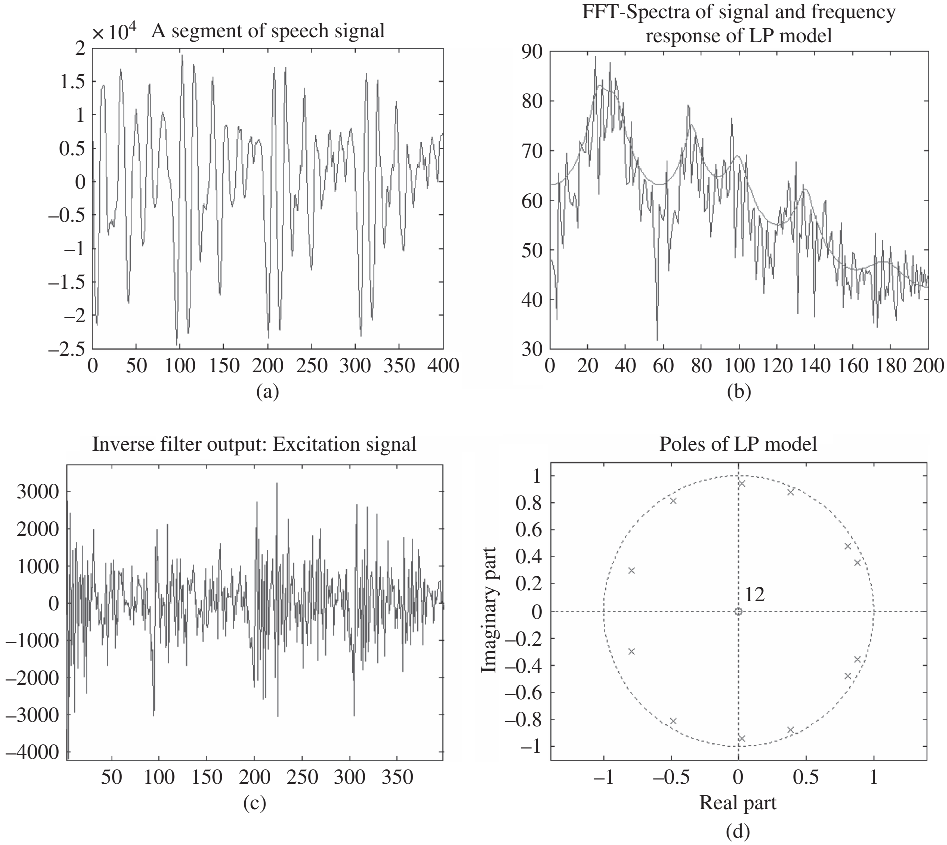


Figure 8.9 (a) A speech segment, (b) FFT-spectrum and LP frequency response of speech superimposed, (c) inverse filter output, (d) the poles of LP model, prediction order $P = 12$. Speech sampling rate = 16 kHz.

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 $x(m-1),\ x(m-2), \ldots,\ x(m-P)$. Similarly, as shown in Figure 8.10, we can define a backward predictor, which predicts a sample $x(m-P)$ from P *future* samples $x(m-P+1), \ldots,\ x(m)$ as

$$\hat{x}(m-P) = \sum_{k=1}^{P} c_k\, x(m-k+1) \tag{8.27}$$

The backward prediction error is defined as the difference between the actual sample and its predicted value:

$$\begin{aligned} b(m) &= x(m-P) - \hat{x}(m-P) \\ &= x(m-P)\ - \sum_{k=1}^{P} c_k\, x(m-k+1) \end{aligned} \tag{8.28}$$

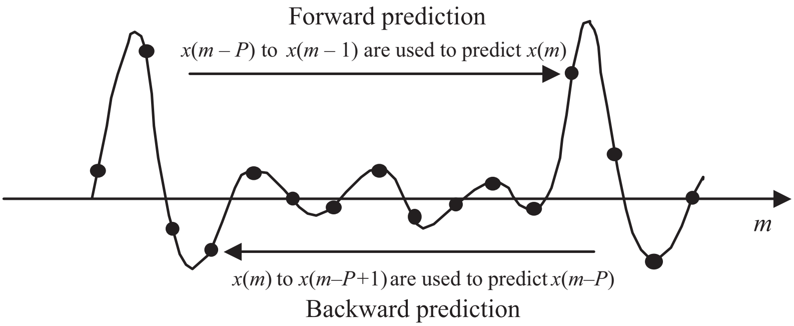

Forward prediction
x(m – P) to x(m – 1) are used to predict x(m)
m
x(m) to x(m–P+1) are used to predict x(m–P)
Backward prediction

Figure 8.10 Illustration of forward and backward predictors.

From Equation (8.28), a signal generated by a backward predictor is given by

$$x(m-P)=\sum_{k=1}^{P} c_k\, x(m-k+1)\ +b(m) \tag{8.29}$$

The coefficients of the least square error backward predictor, obtained by a similar method to that of the forward predictor in Section 8.1.1, are given by

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_P \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \tag{8.30}$$

Note 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 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 form

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_P \\ a_{P-1} \\ a_{P-2} \\ \vdots \\ a_1 \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \tag{8.31}$$

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

$$\boldsymbol{c}=\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_P \end{pmatrix}=\begin{pmatrix} a_P \\ a_{P-1} \\ a_{P-2} \\ \vdots \\ a_1 \end{pmatrix}=\boldsymbol{a}^{\mathrm{B}} \tag{8.32}$$

where the vector $\boldsymbol{a}^{\mathrm{B}}$ is the reversed version of the vector $\boldsymbol{a}$. 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 $[1, \ -a_1, \ \ldots, \ -a_P] = [1, \ -\boldsymbol{a}^{\mathrm{T}}]$. Equations (8.15) and (8.26) may be combined to yield a matrix Equation for the inverse forward predictor coefficients:

$$\begin{pmatrix} r(0) & \boldsymbol{r}_{xx}^{\mathrm{T}} \\ \boldsymbol{r}_{xx} & \boldsymbol{R}_{xx} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a} \end{pmatrix} = \begin{pmatrix} E^{(P)} \\ \boldsymbol{0} \end{pmatrix} \tag{8.33}$$

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

$$\begin{pmatrix} \boldsymbol{R}_{xx} & \boldsymbol{r}_{xx}^{\mathrm{B}} \\ \boldsymbol{r}_{xx}^{\mathrm{BT}} & r(0) \end{pmatrix} \begin{pmatrix} -\boldsymbol{a}^{\mathrm{B}} \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ E^{(P)} \end{pmatrix} \tag{8.34}$$

where $\boldsymbol{r}_{xx}^{\mathrm{T}} = [r_{xx}(1), \cdots, r_{xx}(P)]$ and $\boldsymbol{r}_{xx}^{\mathrm{BT}} = [r_{xx}(P), \cdots, r_{xx}(1)]$. 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

$$\begin{pmatrix} 1 \\ -a_1^{(i)} \\ \vdots \\ -a_{i-1}^{(i)} \\ -a_i^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -a_1^{(i-1)} \\ \vdots \\ -a_{i-1}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -a_{i-1}^{(i-1)} \\ \vdots \\ -a_1^{(i-1)} \\ 1 \end{pmatrix} \tag{8.35}$$

or in more compact vector notation as

$$\begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} \tag{8.36}$$

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, follows 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

$$\begin{pmatrix} -\boldsymbol{a}^{(i)\mathrm{B}} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} + k_i \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} \tag{8.37}$$

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 $\boldsymbol{R}_{\boldsymbol{xx}}^{(i+1)}$ and use the equality

$$\boldsymbol{R}_{\boldsymbol{xx}}^{(i+1)} = \begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{xx}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{xx}^{(i)\mathrm{BT}} & r_{xx}(0) \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & \boldsymbol{r}_{xx}^{(i)\mathrm{T}} \\ \boldsymbol{r}_{xx}^{(i)} & \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} \end{pmatrix} \tag{8.38}$$

to obtain

$$\begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{xx}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{xx}^{(i)\mathrm{BT}} & \boldsymbol{r}_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{xx}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{xx}^{(i)\mathrm{BT}} & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} r_{xx}(0) & \boldsymbol{r}_{xx}^{(i)\mathrm{T}} \\ \boldsymbol{r}_{xx}^{(i)} & \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} \end{pmatrix} \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} \tag{8.39}$$

where in Equation (8.38) and Equation (8.39) $\boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{T}} = [r_{xx}(1), \cdots, r_{xx}(i)]$, and $\boldsymbol{r}_{\boldsymbol{xx}}^{(i)B\mathrm{T}} = [r_{xx}(i), \cdots, r_{xx}(1)]$ is the reversed version of $\boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{T}}$. Matrix–vector multiplication of both sides of Equation (8.39) and the use of Equations (8.33) and (8.34) yields

$$\begin{pmatrix} E^{(i)} \\ \mathbf{0}^{(i)} \end{pmatrix} = \begin{pmatrix} E^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ \Delta^{(i-1)} \end{pmatrix} + k_i \begin{pmatrix} \Delta^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ E^{(i-1)} \end{pmatrix} \tag{8.40}$$

where

$$\begin{aligned} \Delta^{(i-1)} &= \left[1 - \boldsymbol{a}^{(i-1)^T}\right] \boldsymbol{r}_{\boldsymbol{xx}}^{(i)B} \\ &= r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \end{aligned} \tag{8.41}$$

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

$$E^{(i)} = E^{(i-1)} + k_i \Delta^{(i-1)} \tag{8.42}$$

and

$$0 = \Delta^{(i-1)} + k_i \, E^{(i-1)} \tag{8.43}$$

From Equation (8.43),

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \tag{8.44}$$

Substitution of $\boldsymbol{\Delta}^{(i-1)}$ from Equation (8.44) into Equation (8.42) yields

$$\begin{aligned} E^{(i)} &= E^{(i-1)}(1 - k_i^2) \\ &= E^{(0)} \prod_{j=1}^{i} (1 - k_j^2) \end{aligned} \tag{8.45}$$

Note that it can be shown that $\boldsymbol{\Delta}^{(i)}$ is the cross-correlation of the forward and backward prediction errors:

$$\Delta^{(i-1)} = \mathcal{E}[b^{(i-1)}(m-1) e^{(i-1)}(m)] \tag{8.46}$$

The parameter $\boldsymbol{\Delta}^{(i-1)}$ is known as the partial correlation.

8.2.2.1 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 $i = 1, \ldots, P$

$$\Delta^{(i-1)} = r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \tag{8.48}$$

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \tag{8.49}$$

$$a_i^{(i)} = -k_i \tag{8.50}$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i\, a_{i-j}^{(i-1)} \quad 1 \le j \le i-1 \tag{8.51}$$

$$E^{(i)} = (1-k_i^2)\, E^{(i-1)} \tag{8.52}$$

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 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 (8.35) by the input vector $[x(m), x(m-1), \ldots, x(m-i)]$:

$$e^{(i)}(m) = e^{(i-1)}(m) + k_i\, b^{(i-1)}(m-1) \tag{8.53}$$

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 $[x(m-i), x(m-i+1), \ldots, x(m)]$ as

$$b^{(i)}(m) = b^{(i-1)}(m-1) + k_i\, e^{(i-1)}(m) \tag{8.54}$$

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

$$k_i = -\frac{\sum\limits_{m=0}^{N-1} \left(e^{(i-1)}(m)\, b^{(i-1)}(m-1)\right)}{\sum\limits_{m=0}^{N-1} \left(b^{(i-1)}(m-1)\right)^2} \tag{8.55}$$

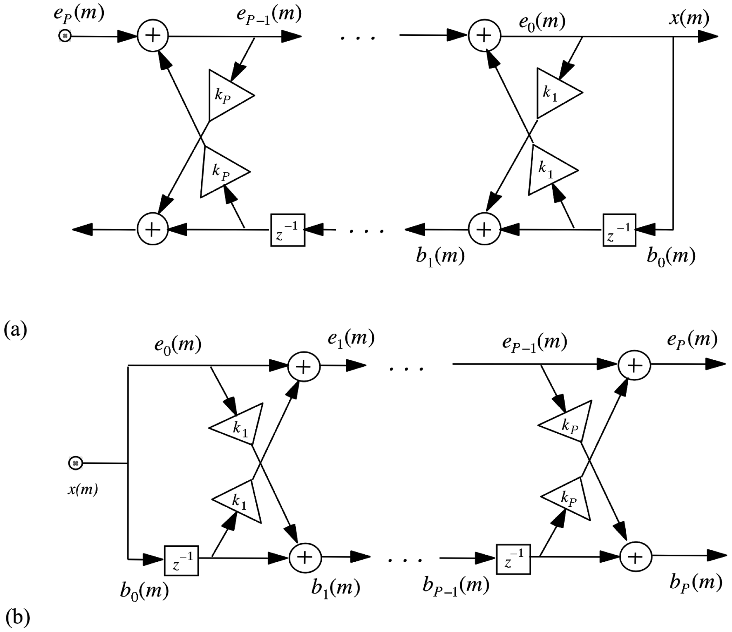


Figure 8.11 (a) Lattice implementation of linear predictor and (b) the inverse lattice linear predictor.

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.

8.2.4.1 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

$$E_{fb}^{(i)} = \sum_{m=0}^{N-1} \left\{ \left[e^{(i)}(m)\right]^2 + \left[b^{(i)}(m)\right]^2 \right\} \tag{8.56}$$

Substitution of Equations (8.53) and (8.54) in Equation (8.56) yields

$$E_{fb}^{(i)} = \sum_{m=0}^{N-1} \left\{ \left(e^{(i-1)}(m) + k_i b^{(i-1)}(m-1)\right)^2 + \left(b^{(i-1)}(m-1) - k_i e^{(i-1)}(m)\right)^2 \right\} \tag{8.57}$$

Minimisation of $E_{fb}^{(i)}$ with respect to the reflection coefficients k_i yields

$$k_i = -\frac{2\sum\limits_{m=0}^{N-1} e^{(i-1)}(m) b^{(i-1)}(m-1)}{\sum\limits_{m=0}^{N-1} \left(b^{(i-1)}(m-1)\right)^2 + \left(e^{(i-1)}(m)\right)^2} \tag{8.58}$$

8.2.5 Simultaneous Minimisation of the Backward and Forward Prediction Errors

From Equation (8.32) we have that 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

$$\begin{aligned} E_{fb}^{(P)} &= \sum_{m=0}^{N-1} \left\{ \left[e^{(P)}(m)\right]^2 + \left[b^{(P)}(m)\right]^2 \right\} \\ &= \sum_{m=0}^{N-1} \left\{ \left[x(m) - \sum_{k=1}^{P} a_k x(m-k) \right]^2 + \left[x(m-P) - \sum_{k=1}^{P} a_k x(m-P+k) \right]^2 \right\} \\ &= (\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a})^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}) + \left(\boldsymbol{x}^{\mathrm{B}} - \boldsymbol{X}^{\mathrm{B}}\boldsymbol{a}\right)^{\mathrm{T}} \left(\boldsymbol{x}^{\mathrm{B}} - \boldsymbol{X}^{\mathrm{B}}\boldsymbol{a}\right) \end{aligned} \tag{8.59}$$

where $\boldsymbol{X}$ and $\boldsymbol{x}$ are the signal matrix and vector defined by Equations (8.16) and (8.17), and similarly $\boldsymbol{X}^{\mathrm{B}}$ and $\boldsymbol{x}^{\mathrm{B}}$ 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

$$\boldsymbol{a} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{X}^{\mathrm{BT}}\boldsymbol{X}^{\mathrm{B}}\right)^{-1} \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{X}^{\mathrm{BT}}\boldsymbol{x}^{\mathrm{B}}\right) \tag{8.60}$$

where $\boldsymbol{X}^{\mathrm{BT}}$ is the transpose of $\boldsymbol{X}^{\mathrm{B}}$. Note that for an ergodic signal as the signal length N increases Equation (8.60) converges to the so-called normal Equation (8.14).

8.2.6 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

$$\Delta E^{(i)} = E^{(i-1)} - E^{(i)} \tag{8.61}$$

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.

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.

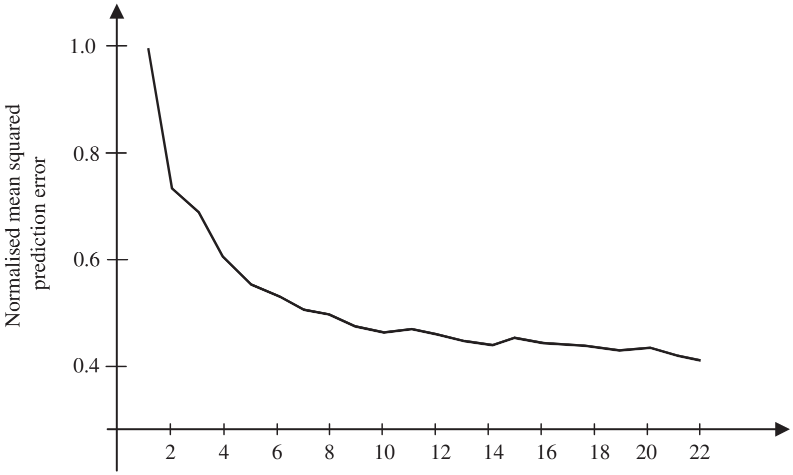


Figure 8.12 Illustration of the decrease in the normalised mean squared prediction error with the increasing predictor length for a speech signal.

When the model order is less than the correct order, the signal is under-modelled. In this case the prediction error is not well de-correlated and will be more than the optimal minimum. A further consequence of under-modelling 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 over-modelled. An over-modelled 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-Term and Long-Term Predictors

For quasi-periodic signals, such as voiced speech, two types of correlation structures can be utilised for a more accurate prediction:

(1) The short-term correlation, which is the correlation of each sample with the P immediate past samples: $x(m-1), \ldots, x(m-P)$.
(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: $x(m-T+Q), \ldots, x(m-T-Q)$.

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 correlation of a signal may be modelled by the linear prediction 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

$$\hat{e}(m) = \sum_{k=-Q}^{Q} p_k \, e(m-T-k) \tag{8.62}$$

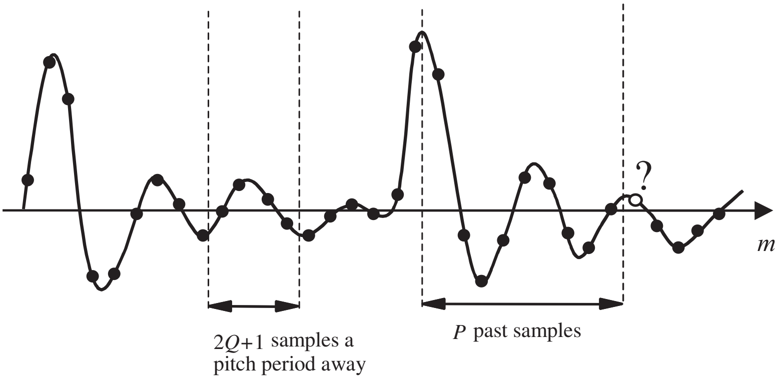


Figure 8.13 Illustration of the short-term relation of a sample with the P immediate past samples and the long-term relation with the samples a pitch period away.

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 non-zero 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 $\mathrm{E}[e^2(m)]$ 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

$$x(m) = \underbrace{\sum_{k=1}^{P} a_k\, x(m-k)}_{\text{short-term prediction}} + \underbrace{\sum_{k=-Q}^{Q} p_k\, x(m-k-T) + \varepsilon(m)}_{\text{long-term prediction}} \tag{8.65}$$

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 $\mathrm{E}[e^2(m)]$ results in the following solution for the pitch predictor:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{+Q} \end{pmatrix} = \begin{pmatrix} r(0) & r(1) & \cdots & r(P-1) & r(T-Q-1) & r(T-Q) & \cdots & r(T+Q-1) \\ r(1) & r(0) & \cdots & r(P-2) & r(T-Q-2) & r(T-Q-1) & \cdots & r(T+Q-2) \\ r(2) & r(1) & \cdots & r(P-3) & r(T-Q-3) & r(T+Q-2) & \cdots & r(T+Q-3) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \cdots & r(0) & r(T-Q-P) & r(T-Q-P+1) & \cdots & r(T+Q-P) \\ r(T-Q-1) & r(T-Q-2) & \cdots & r(T-Q-P) & r(0) & r(1) & \cdots & r(2Q) \\ r(T-Q) & r(T-Q-1) & \cdots & r(T-Q-P+1) & r(1) & r(0) & \cdots & r(2Q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(T+Q-1) & r(T+Q-2) & \cdots & r(T+Q-P) & r(2Q) & r(2Q-1) & \cdots & r(0) \end{pmatrix}^{-1} \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T-Q) \\ r(T-Q+1) \\ \vdots \\ r(T+Q) \end{pmatrix} \tag{8.66}$$

In Equation (8.66), for simplicity the subscript xx of $r_{xx}(k)$ has been omitted.

8.4 MAP Estimation of Predictor Coefficients

The posterior probability density function of a predictor coefficient vector $\boldsymbol{a}$, given a signal $\boldsymbol{x}$ and the initial samples $\boldsymbol{x}_{\mathbf{I}}$, can be expressed, using Bayes' rule, as

$$f_{A|X,X_{\mathbf{I}}}(\boldsymbol{a}|\boldsymbol{x},\boldsymbol{x}_{\mathbf{I}}) = \frac{f_{X|A,X_I}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_{\mathbf{I}}) f_{A|X_{\mathbf{I}}}(\boldsymbol{a}|\boldsymbol{x}_{\mathbf{I}})}{f_{X|X_I}(\boldsymbol{x}|\boldsymbol{x}_{\mathbf{I}})} \tag{8.67}$$

In Equation (8.67), the pdfs are conditioned on P initial signal samples $\boldsymbol{x}_{\mathbf{I}} = [x(-P), x(-P+1), \ldots, x(-1)]$. Note that for a given set of samples $[\boldsymbol{x}, \boldsymbol{x}_{\mathbf{I}}]$, $f_{X|X_I}(\boldsymbol{x}|\boldsymbol{x}_I)$ is a constant, and it is reasonable to assume that $f_{A|X_I}(\boldsymbol{a}|\boldsymbol{x}_{\mathbf{I}}) = f_A(\boldsymbol{a})$.

8.4.1 Probability Density Function of Predictor Output

The pdf $f_{X|A,X_{\mathrm{I}}}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_{\mathrm{I}})$ of the signal $\boldsymbol{x}$, given the predictor coefficient vector $\boldsymbol{a}$ and the initial samples $\boldsymbol{x}_{\mathrm{I}}$, is equal to the pdf of the input signal $\boldsymbol{e}$:

$$f_{X|A,X_I}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_I) = f_E(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}) \tag{8.68}$$

where the input signal vector is given by

$$\boldsymbol{e} = -\boldsymbol{X}\boldsymbol{a} \tag{8.69}$$

and $f_E(\boldsymbol{e})$ is the pdf of $\boldsymbol{e}$. 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

$$\begin{aligned} f_{X|A,X_\mathrm{I}}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_\mathrm{I}) &= f_E(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a}) \\ &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(\frac{1}{2\sigma_e^2}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})^\mathrm{T}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})\right) \end{aligned} \tag{8.71}$$

An alternative form of Equation (8.71) can be obtained by rewriting Equation (8.70) in the 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 compact notation Equation (8.72) can be written as

$$\boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} \tag{8.73}$$

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

$$f_{X|A,X_I}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_\mathrm{I}) = \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^\mathrm{T}\boldsymbol{A}^\mathrm{T}\boldsymbol{A}\boldsymbol{x}\right) \tag{8.74}$$

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 $\boldsymbol{\mu}_a$ and a covariance matrix $\boldsymbol{\Sigma}_{aa}$:

$$f_A(\boldsymbol{a}) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{aa}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{a}-\boldsymbol{\mu}_a)^\mathrm{T}\boldsymbol{\Sigma}_{aa}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_a)\right) \tag{8.75}$$

Substituting Equations (8.72) and (8.76) in Equation (8.67), the posterior pdf of the predictor coefficient vector $f_{A|X,X_\mathrm{I}}(\boldsymbol{a}|\boldsymbol{x},\boldsymbol{x}_\mathrm{I})$ can be expressed as

$$\begin{aligned} f_{A|X,X_I}(\boldsymbol{a}|\boldsymbol{x},\boldsymbol{x}_\mathrm{I}) &= \frac{1}{f_{X|X_\mathrm{I}}(\boldsymbol{x}|\boldsymbol{x}_\mathrm{I})} \frac{1}{(2\pi)^{(N+P)/2}\sigma_e^N|\boldsymbol{\Sigma}_{aa}|^{1/2}} \\ &\quad \times \exp\left\{-\frac{1}{2}\left[\frac{1}{\sigma_e^2}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})^\mathrm{T}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a}) + (\boldsymbol{a}-\boldsymbol{\mu}_a)^\mathrm{T}\boldsymbol{\Sigma}_{aa}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_a)\right]\right\} \end{aligned} \tag{8.76}$$

The maximum a posteriori estimate is obtained by maximising the log-likelihood function:

$$\frac{\partial}{\partial \boldsymbol{a}}\left[\ln f_{A|X,X_I}\left(\boldsymbol{a}|\boldsymbol{x},\boldsymbol{x}_I\right)\right]=\frac{\partial}{\partial \boldsymbol{a}}\left[\frac{1}{\sigma_e^2}\left(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a}\right)^T\left(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a}\right)+\left(\boldsymbol{a}-\boldsymbol{\mu}_a\right)^T\boldsymbol{\Sigma}_{aa}^{-1}\left(\boldsymbol{a}\;-\boldsymbol{\mu}_a\right)\right]=0 \tag{8.77}$$

This yields

$$\hat{\boldsymbol{a}}^{\mathrm{MAP}}=\left(\boldsymbol{\Sigma}_{aa}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{\Sigma}_{aa}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}+\sigma_e^2\left(\boldsymbol{\Sigma}_{aa}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{\mu}_a \tag{8.78}$$

Note that as the Gaussian prior tends to a uniform prior, the determinant covariance matrix $\boldsymbol{\Sigma}_{aa}$ of the Gaussian prior increases, and the MAP solution tends to the least square error solution

$$\hat{\boldsymbol{a}}^{\mathrm{LS}}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}\right) \tag{8.79}$$

Similarly as the observation length N increases the signal matrix term $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ becomes more significant than $\boldsymbol{\Sigma}_{aa}$ and again the MAP solution tends to a least squared 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

$$X(z)=G(z)\;V(z)\;L(z) \tag{8.80}$$

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

$$V(z,m)=G(m)\prod_{k=1}^{N}\frac{1}{1-2r_k\varphi_k(m)z^{-1}+r_k^2(m)z^{-2}} \tag{8.81}$$

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. The relationship between poles of linear prediction models and formants is not one-to-one. 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 characterised by a parameter vector comprising of the frequency F_k, bandwidth B_k and magnitude of the resonance M_k and their temporal slope of variation as

$$F_k=[F_k,B_k,M_k,\Delta F_k,\Delta B_k,\Delta M_k]\quad k=1,\ldots,M \tag{8.82}$$

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

$$\Delta F_k(t)=\frac{\sum\limits_{m=1}^{N}m\big(F_k(t+m)-F_k(t-m)\big)}{\sum\limits_{m=1}^{N}2m^2}\qquad k=1,\ldots,M \tag{8.83}$$

There are three issues in modelling formants: (1) modelling the distribution of formants using a probability model, (2) tracking 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.

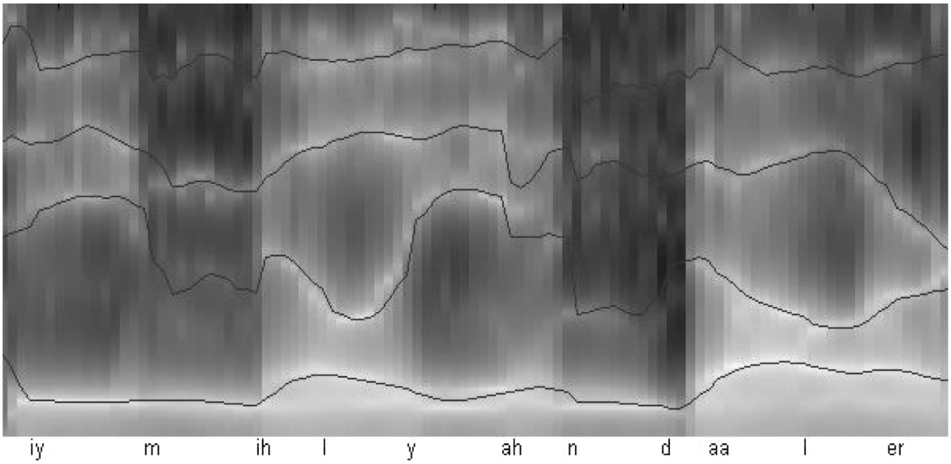


Figure 8.14 An example of formant tracks superimposed on an LP spectrogram. Lighter regions on the spectrogram correspond to high energy.

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 minimise 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 lower-order linear prediction model as shown in Figure 8.15.

The advantages of using a sub-band LP model are:

(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 solutions of the sub-band LP models require the inversion of a number of smaller correlation matrices with better numerical stability properties. For example, 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 $x_k(m),\ k=1,\ldots,N$. The k^{th} sub-band signal is modelled using a low-order linear prediction model as

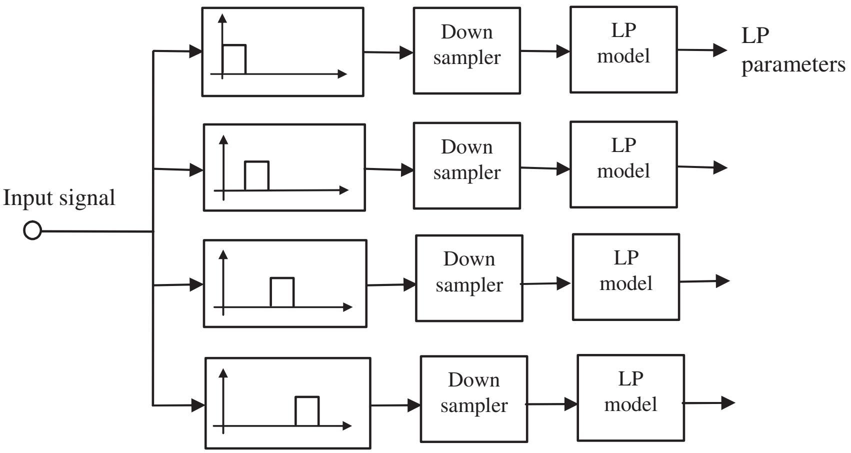


Figure 8.15 Configuration of a sub-band linear prediction (LP) model.

$$x_k(m) = \sum_{i=1}^{P_k} a_k(i)x_k(m-i) + g_k e_k(m) \tag{8.84}$$

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

$$P_{EE}(f,k) = \begin{cases} 1 & f_{k,\text{start}} < f < f_{k,\text{end}} \\ 0 & \text{otherwise} \end{cases} \tag{8.85}$$

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

$$r_{ee}(m) = B_k \operatorname{sinc}\left[m(B_k - f_{k_0})/2\right] \tag{8.86}$$

where B_k and f_{k_0} are the bandwidth and the centre frequency of the k^{th} sub-band respectively. To ensure that sub-band LP parameters only model the signal within that sub-band (and do not model the bandpass filters) 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

$$\begin{aligned} y(m) &= x(m) + \; n(m) \\ &= \sum_{k=1}^{P} a_k\, x(m-k) \; + e(m) + n(m) \end{aligned} \tag{8.87}$$

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

$$\boldsymbol{R}_{yy}\hat{\boldsymbol{a}} = \boldsymbol{r}_{yy} \tag{8.88}$$

where $\boldsymbol{R}_{yy}$ and $\boldsymbol{r}_{yy}$ are the autocorrelation matrix and vector of the noisy signal $y(m)$. For an additive noise model, Equation (8.88) can be written as

$$(\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})\,(\boldsymbol{a} + \tilde{\boldsymbol{a}}) = (\boldsymbol{r}_{xx} + \boldsymbol{r}_{nn}) \tag{8.89}$$

where $\tilde{\boldsymbol{a}}$ is the error in the predictor coefficient 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' rule, the MAP estimate of the predictor coefficient vector $\boldsymbol{a}$, given an observation signal vector $\boldsymbol{y} = [y(0),\ y(1),\ \ldots,\ y(N-1)]$, and the initial samples vector $\boldsymbol{x}_{\mathrm{I}}$ is

$$f_{A|Y,X_I}(\boldsymbol{a}|\boldsymbol{y},\boldsymbol{x}_{\mathrm{I}}) = \frac{f_{Y|A,X_{\mathrm{I}}}(\boldsymbol{y}|\boldsymbol{a},\boldsymbol{x}_{\mathrm{I}})f_{A,X_{\mathrm{I}}}(\boldsymbol{a},\boldsymbol{x}_{\mathrm{I}})}{f_{Y,X_{\mathrm{I}}}(\boldsymbol{y},\boldsymbol{x}_{\mathrm{I}})} \tag{8.90}$$

Consider the variance of the signal $\boldsymbol{y}$ in the argument of the term $f_{Y|A,X_I}(\boldsymbol{y}|\boldsymbol{a},\boldsymbol{x}_I)$ in Equation (8.90). The innovation (i.e. prediction error) of $y(m)$ can be defined as

$$\begin{aligned} \varepsilon(m) &= y(m) - \sum_{k=1}^{P} \hat{a}_k\, y(m-k) \\ &= e(m) + \varepsilon(m) + n(m) - \sum_{k=1}^{P} \hat{a}_k\, n(m-k) \end{aligned} \tag{8.91}$$

The variance of $y(m)$, given the previous P samples and the coefficient vector $\hat{\boldsymbol{a}}$, is the variance of the innovation signal given by

$$\mathrm{Var}\left[y(m) \mid y(m-1),\ \ldots,\ y(m-P),\hat{\boldsymbol{a}}\right] = \sigma_e^2 + \sigma_\varepsilon^2 + \sigma_n^2 - \sigma_n^2 \sum_{k=1}^{P} \hat{a}_k^2 \tag{8.92}$$

where $\sigma_e^2, \sigma_\varepsilon^2$ 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 $[y(m)|\, y(m-1),\ \ldots,\ y(m-P),\ \hat{\boldsymbol{a}}]$ is a function of the coefficient vector $\hat{\boldsymbol{a}}$. Consequently, maximisation of $f_{Y|A,X_{\mathrm{I}}}(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x}_{\mathrm{I}})$ with respect to the vector $\hat{\boldsymbol{a}}$ is a non-linear and non-trivial exercise.

Lim and Oppenheim proposed the following iterative process where an estimate $\hat{\boldsymbol{a}}$ of the predictor coefficient vector is used to make an estimate $\hat{\boldsymbol{x}}$ of the signal vector, and the signal estimate $\hat{\boldsymbol{x}}$ is then used to improve the estimate of the parameter vector $\hat{\boldsymbol{a}}$, and the process is iterated until convergence. The posterior pdf of the noise-free signal $\boldsymbol{x}$ given the noisy signal $\boldsymbol{y}$ and an estimate of the parameter vector $\hat{\boldsymbol{a}}$ is given by

$$f_{X|A,Y}\left(\boldsymbol{x}|\hat{\boldsymbol{a}},\boldsymbol{y}\right)=\frac{f_{Y|A,X}\left(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x}\right)f_{X|A}\left(\boldsymbol{x}|\hat{\boldsymbol{a}}\right)}{f_{Y|A}\left(\boldsymbol{y}|\hat{\boldsymbol{a}}\right)} \tag{8.93}$$

Consider the likelihood term $f_{Y|A,X}(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x})$. Since the noise is additive, we have

$$\begin{aligned} f_{Y|A,X}\left(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x}\right) &= f_N\left(\boldsymbol{y}-\boldsymbol{x}\right) \\ &= \frac{1}{\left(2\pi\sigma_n^2\right)^{N/2}}\exp\left[-\frac{1}{2\sigma_n^2}\left(\boldsymbol{y}-\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{x}\right)\right] \end{aligned} \tag{8.94}$$

Assuming that the input of the predictor model is a zero-mean Gaussian process with variance σ_e^2, the pdf of the signal $\boldsymbol{x}$ given an estimate of the predictor coefficient vector $\boldsymbol{a}$ is

$$\begin{aligned} f_{Y|A,X}\left(\boldsymbol{x}|\hat{\boldsymbol{a}}\right) &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}}\exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}\right) \\ &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}}\exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}\right) \end{aligned} \tag{8.95}$$

where $\boldsymbol{e}=\hat{\boldsymbol{A}}\boldsymbol{x}$ as in Equation (8.73). Substitution of Equations (8.94) and (8.95) in Equation (8.93) yields

$$f_{X|A,Y}\left(\boldsymbol{x}|\hat{\boldsymbol{a}},\boldsymbol{y}\right)=\frac{1}{f_{Y|A}\left(\boldsymbol{y}|\hat{\boldsymbol{a}}\right)}\frac{1}{\left(2\pi\sigma_n\sigma_e\right)^N}\exp\left(-\frac{1}{2\sigma_n^2}\left(\boldsymbol{y}-\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{x}\right)-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}\right) \tag{8.96}$$

In Equation (8.92), for a given signal $\boldsymbol{y}$ and coefficient vector $\hat{\boldsymbol{a}}$, $f_{Y|A}(\boldsymbol{y}|\hat{\boldsymbol{a}})$ is a constant. From Equation (8.92), the ML signal estimate is obtained by maximising the log-likelihood function as

$$\frac{\partial}{\partial\boldsymbol{a}}\left(\ln f_{X|A,Y}\left(\boldsymbol{x}|\hat{\boldsymbol{a}},\boldsymbol{y}\right)\right)=\frac{\partial}{\partial\boldsymbol{x}}\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}-\frac{1}{2\sigma_n^2}\left(\boldsymbol{y}-\boldsymbol{x}\right)^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{x}\right)\right)=\boldsymbol{0} \tag{8.97}$$

which gives

$$\hat{\boldsymbol{x}}=\sigma_e^2\left(\sigma_n^2\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{y} \tag{8.98}$$

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 $\boldsymbol{a}$ is given by

$$\hat{\boldsymbol{a}}(\hat{\boldsymbol{x}})=\left(\hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{X}}\right)^{-1}\left(\hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{x}}\right) \tag{8.99}$$

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 $\hat{\boldsymbol{x}}_0=\boldsymbol{y}$

For iterations $i = 0,\ 1,\ \ldots$

Step 1: Estimate the predictor parameter vector $\hat{\boldsymbol{a}}_i$:

$$\hat{\boldsymbol{a}}_i(\hat{\boldsymbol{x}}_i) = \left(\hat{\boldsymbol{X}}_i^{\mathrm{T}}\hat{\boldsymbol{X}}_i\right)^{-1}\left(\hat{\boldsymbol{X}}_i^{\mathrm{T}}\hat{\boldsymbol{x}}_i\right) \tag{8.100}$$

Step 2: Calculate an estimate of the model gain G using Parseval's theorem:

$$\frac{1}{N}\sum_{f=0}^{N-1}\frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}\,e^{-j2\pi fk/N}\right|^2} = \sum_{m=0}^{N-1}y^2(m) - N\hat{\sigma}_n^2 \tag{8.101}$$

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:

$$\hat{P}_{X_iX_i}(f) = \frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}\,e^{-j2\pi fk/N}\right|^2} \tag{8.102}$$

Step 4: Calculate the Wiener filter frequency response:

$$\hat{W}_i(f) = \frac{\hat{P}_{X_iX_i}(f)}{\hat{P}_{X_iX_i}(f) + \hat{P}_{N_iN_i}(f)} \tag{8.103}$$

where $\hat{P}_{N_iN_i}(f) = \hat{\sigma}_n^2$ is an estimate of the noise power spectrum.

Step 5: Filter the magnitude spectrum of the noisy speech as

$$\hat{X}_{i+1}(f) = \hat{W}_i(f)Y(f) \tag{8.104}$$

Restore the time domain signal $\hat{\boldsymbol{x}}_{i+1}$ by combining $\hat{X}_{i+1}(f)$ with the phase of noisy signal and the complex signal to time domain.

Step 6: Go to 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.

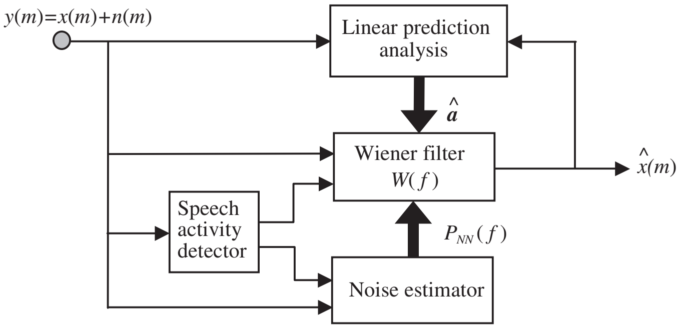


Figure 8.16 Iterative signal restoration based on linear prediction model of speech.

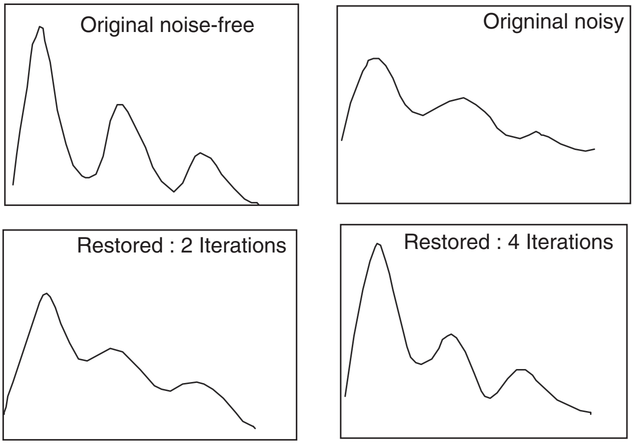


Figure 8.17 Illustration of restoration of a noisy signal with iterative linear prediction based method.

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

$$y_k(m) = x_k(m) + n_k(m) \tag{8.105}$$

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

$$\begin{aligned} W_k(f) &= \frac{P_{X,k}(f)}{P_{Y,k}(f)} \\ &= \frac{g_{X,k}^2}{\left|A_{X,k}(f)\right|^2} \frac{\left|A_{Y,k}(f)\right|^2}{g_{Y,k}^2} \end{aligned} \tag{8.106}$$

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} sub-band respectively. From Equation (8.112) the square-root Wiener filter is given by

$$W_k^{1/2}(f) = \frac{g_{X,k}}{\left|A_{X,k}(f)\right|} \frac{\left|A_{Y,k}(f)\right|}{g_{Y,k}} \tag{8.107}$$

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):

$$z_k(m) = \sum_{i=1}^{P} a_{Xk}(i) z_k(m-i) + \frac{g_X}{g_Y} y_k(m) \tag{8.108}$$

$$\hat{x}_k(m) = \sum_{i=0}^{P} a_{Yk}(i) z_k(m-i) \tag{8.109}$$

where $\hat{x}_k(m)$ is the restored estimate of the clean speech signal $x_k(m)$, $z_k(m)$ is an intermediate signal, $a_{Yk}(i)$ are the coefficients of the linear predictor model of the noisy signal for the k^{th} sub-band and $a_{Xk}(i)$ are an estimate of the coefficients of the linear prediction model of the k^{th} sub-band of clean speech.

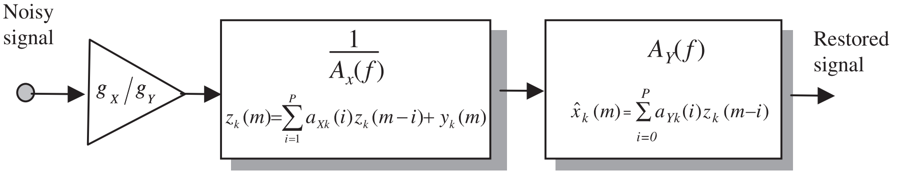

Noisy signal
g_X/g_Y
$\frac{1}{A_x(f)}$
$z_k(m)=\sum_{i=1}^{P} a_{Xk}(i)z_k(m-i)+ y_k(m)$
$A_Y(f)$
$\hat{x}_k(m)=\sum_{i=0}^{P} a_{Yk}(i)z_k(m-i)$
Restored signal

Figure 8.18 A cascade implementation of the LP squared-root Wiener filter.

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 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 quasi-periodic 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. Section 8.5 introduced formant tacking linear predictors which are used in Chapter 17 for speech enhancement. In Section 8.6, the sub-band linear prediction method was formulated. Finally in Section 8.7, a linear prediction model was applied to the restoration of a signal observed in additive noise.

Bibliography

Akaike H. (1970) Statistical Predictor Identification, *Annals of the Institute of Statistical Mathematics*. **22**: 203–217.

Akaike H. (1974) A New Look at Statistical Model Identification, *IEEE Trans. Automatic Control*, **AC-19**: 716–723, Dec.

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.

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. PhD thesis, Stanford University, Stanford, CA.

Cohen J. and Cohen, P. (1975) *Applied Multiple Regression/Correlation Analysis for the Behavioural Sciences*. Halsted, New York.

Draper N.R. and Smith, H. (1981). *Applied Regression Analysis*, 2nd edn, John Wiley & Sons, Inc, New York.

Durbin J. (1959) Efficient Estimation of Parameters in Moving Average Models. *Biometrica*, **46**: 306–318.

Durbin J. (1960) The Fitting of Time Series Models. *Rev. Int. Stat. Inst.*, **28**: 233–244.

Fuller W.A (1976) *Introduction to Statistical Time Series*. John Wiley & Sons, Inc, New York.

Hansen J.H. and Clements M.A. (1987) Iterative Speech Enhancement with Spectral Constrains. *IEEE Proc. Int. Conf. on Acoustics, Speech and Signal Processing* **ICASSP-87, 1**: 189–192, Dallas, April.

Hansen J.H. and Clements M.A. (1988) Constrained Iterative Speech Enhancement with Application to Automatic Speech Recognition. *IEEE Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, **ICASSP-88, 1**: 561–564, New York, April.

Hocking R.R. (1996) *The Analysis of Linear Models*. John Wiley & Sons.

Kobatake H., Inari J. and Kakuta S. (1978) Linear Prediction Coding of Speech Signals in a High Ambient Noise Environment. *IEEE Proc. Int. Conf. on Acoustics, Speech and Signal Processing*: 472–475, April.

Lim J.S. and Oppenheim A.V. (1978) All-Pole Modelling of Degraded Speech. *IEEE Trans. Acoustics, Speech and Signal Processing*, **ASSP-26, 3**: 197–210, June.

Lim J.S. and Oppenheim A.V. (1979) Enhancement and Bandwidth Compression of Noisy Speech, *Proc. IEEE*, **67**: 1586–1604.

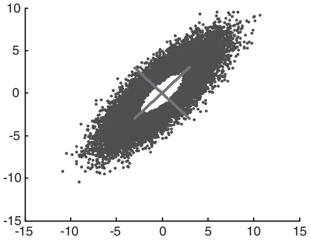
Makoul J. (1975) Linear Prediction: A Tutorial Review. *Proc. IEEE*, **63**: 561–580.

Markel J.D. and Gray A.H. (1976) *Linear Prediction of Speech*. Springer Verlag, New York.

Rabiner L.R. and Schafer R.W. (1976) *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ.

Stockham T.G., Cannon T.M. and Ingebretsen R.B (1975) Blind Deconvolutiopn Through Digital Signal Processing. *IEEE Proc*., **63**, 4: 678–692.

Tong H. (1975) Autoregressive Model Fitting with Noisy Data by Akaike's Information Criterion, *IEEE Trans. Information Theory*, **IT-23**: 409–448.



9

Eigenvalue Analysis and Principal Component Analysis

Eigen is a German word that translates as ‘own’, ‘peculiar to’, ‘characteristic’ or ‘individual’. To understand eigen vectors we need to understand the basic operational function of a matrix. In general, linear matrices change the directions and the magnitudes of the vectors on which they operate. Eigen vectors of a matrix are those vectors whose directions are unaffected by a transformation through the matrix. It is an interesting property of a matrix that if a vector is repeatedly passed through a matrix it eventually aligns itself with the most significant eigen vector of the matrix.

The eigen analysis of a matrix finds a set of characteristic orthonormal eigen vectors and their magnitudes called eigenvalues. A matrix can be decomposed and expressed in terms of its eigen vectors and eigenvalues. Similarly, matrix operations can be expressed in terms of operations on the eigen vectors and eigenvalues.

Eigen analysis is useful in signal processing applications such as the diagonalisation of correlation matrices, adaptive filtering, radar signal processing, feature extraction, pattern recognition, signal coding, model order estimation, noise estimation and separation of mixed biomedical or communication signals.

A major application of eigen analysis is in analysis of the covariance matrix of a signal a process known as the principal component analysis (PCA). PCA is widely used for feature extraction and dimension reduction by disposing of those orthogonal dimensions or features that have insignificant variance or very low signal to noise ratio. PCA is essentially eigen analysis of the covariance matrix of a random process. PCA is the optimal feature transform for Gaussian processes which do not possess higher than second order statistics (i.e. covariance statistic). PCA allows the transformation and representation of a signal in terms of the coefficients of a set of orthonormal eigen vectors, such that the principal components (i.e. the most significant components) correspond to the principal dimensions that have the largest eigenvalues (or variances) and are most significant.

9.1 Introduction – Linear Systems and Eigen Analysis

A common requirement in signal processing is the linear transformation of an M-dimensional signal vector $\boldsymbol{x}$ to an N-dimensional output signal vector $\boldsymbol{y}$, via an $N \times M$ dimensional matrix system $\boldsymbol{A}$, as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{9.1}$$

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

The output vector $\boldsymbol{y}$ can be expressed as a weighted combination of the column vectors of $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_M]$ as

$$\boldsymbol{y} = \boldsymbol{a}_1 x_1 + \boldsymbol{a}_2 x_2 + \cdots + \boldsymbol{a}_N x_N \tag{9.2}$$

where $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iN}]^T$ is the N-dimensional i^{th} column of $\boldsymbol{A}$.

Note that all linear filtering and signal transformation and filtering methods can be cast in the form of Equation (9.1). The matrix $\boldsymbol{A}$ is the system that transforms the input signal vector $\boldsymbol{x}$ to the output signal vector $\boldsymbol{y}$, hence the analysis of the structure of $\boldsymbol{A}$ and development of methods for efficient representation and application of $\boldsymbol{A}$ are of particular interest in signal processing and system analysis.

Eigen analysis is a method of representation of a linear system, such as a covariance matrix, in terms of a set of orthonormal eigen vectors and the corresponding eigenvalues. The eigenvalues represent the variance or power of the process along the corresponding eigen vectors. The bigger an eigenvalue the more significant the corresponding eigen vector. The core idea is that the orthonormal eigen components lend themselves to independent processing and easier manipulation and interpretation.

The main advantages of eigen analysis can be listed as follows:

(1) The eigenvectors of a matrix form a set of orthonormal basis vectors in terms of which the matrix can be expressed and applied.
(2) Eigen analysis is used in principal component analysis for feature extraction in coding and pattern recognition in applications such as speech, image processing and radar signal processing.
(3) The eigen vectors of the covariance matrix of a process can be used to transform and pre-whiten the process and diagonalise its covariance matrix. Pre-whitening is useful for many applications such as for faster convergence of adaptive filters in echo cancellation, for ease of calculation of the inverse of covariance matrix often required in probability estimation and in independent component analysis.
(4) Eigenvalues can be used for estimation of the dimensionality of a linear system. It is possible to reduce the dimensions of a matrix or a process to its most significant eigenvalues and eigen vectors.
(5) For white noise, when the dimensions of the covariance matrix of a random process exceeds the required dimension for modelling the signal (e.g. speech or image), then the smallest eigenvalues correspond to the noise power.

9.1.1 A Geometric Interpretation of Eigenvalues and Eigenvectors

Before a formal introduction to eigen analysis in Section 9.2, here, the following simple examples provide a useful geometric interpretation and visualisation of eigenvectors.

Example 9.1 Eigen vectors of a 'Flip' matrix

Figure 9.1 illustrates a very simple matrix transformation that flips a vector [x y] about the y-axis: the transformation changes the sign of the x-component of the vector. It is clear that the y-axis, $[0\ 1]^T$, is an eigen vector of this transformation, as the direction of the vector [0 y] is not affected by the transformation; $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ y \end{bmatrix} = y \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, the x-axis $[1\ 0]^T$ is also an eign vector with a negative eigenvalue since the direction of x-axis is flipped 180° by the transformation; $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 0 \end{bmatrix} = -x \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Example 9.2 Eigen vectors of matrix $A = \begin{bmatrix} \mathbf{1} & \mathbf{0.5} \\ \mathbf{0.5} & \mathbf{1} \end{bmatrix}$ and transformation of a circle by A

Figure (9.2) shows the result of transformation of a circle by a matrix $\boldsymbol{A}$. The transformation changes the shape of the circle and also rotates the points on the circle. For example, after transformation, the point

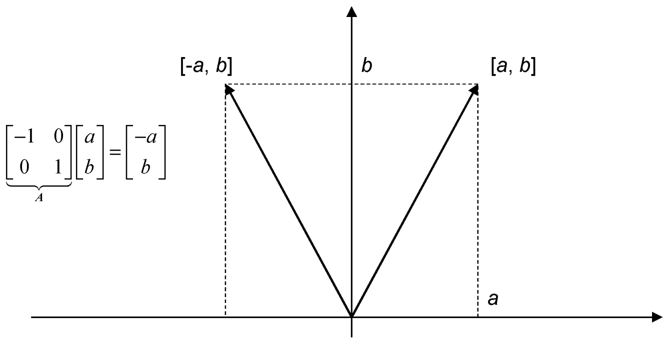

$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}}_{A}\begin{bmatrix} a \\ b \end{bmatrix}=\begin{bmatrix} -a \\ b \end{bmatrix}$
[-a, b]
b
[a, b]
a

Figure 9.1 Illustration of transformation of a vector. In this simple example the vertical component of the vector is unaffected hence the vector $[0\ 1]^T$ is an eigen vector of the matrix $\boldsymbol{A}$.

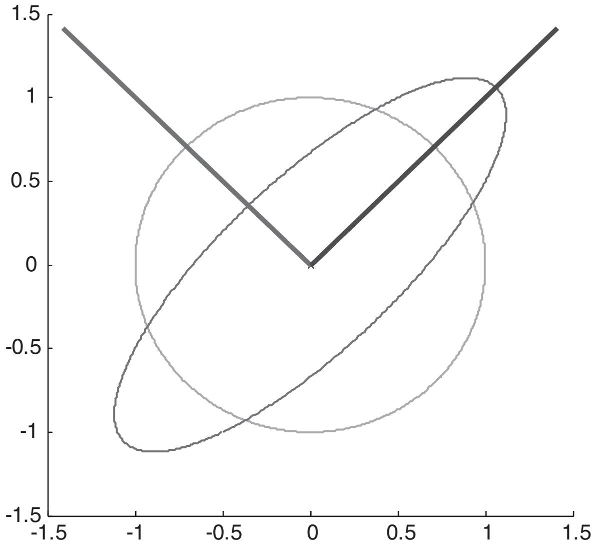


Figure 9.2 Circle is transformed (into an ellipse shape) by the matrix $\boldsymbol{A}=\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$. The eigen vectors of the matrix are the principal dimensions of the transformed curve. The eigen vectors are $[0.7071\ 0.7071]^T$ and $[-0.7071\ 0.7071]^T$.

$[x=0, y=1]$ on the circle becomes $[x=0.5, y=1]$. The eigen vectors of the matrix are superimposed on the diagram. Note that after the transformation the rotated ellipse-shaped curve is positioned with its principal directions in the direction of the eigenvectors of the transformation matrix $\boldsymbol{A}$ as illustrated.

Example 9.3 Eigen vectors of $A=\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ and transformation of a Random Process by matrix A

Figure 9.3(a) shows the scatter diagram for a two-dimensional random process. Figure 9.3(b) shows the process after a linear transformation by a matrix $\boldsymbol{A}$. The values of the individual coefficients, a_{ij}, of the matrix $\boldsymbol{A}$ are also shown. In Figure 9.3(b) the eigen vectors of the transformation matrix $\boldsymbol{A}$ are

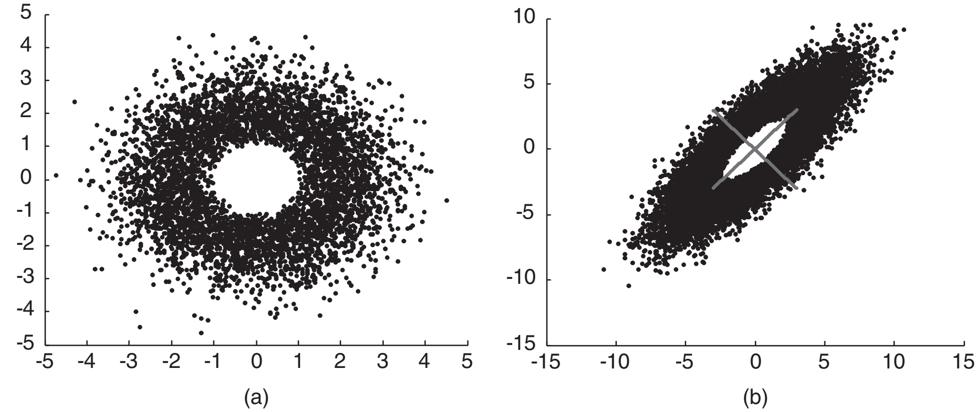


Figure 9.3 (a) A two-dimensional random process, (b) after transformation by the matrix $\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$. Note in (b) that the eigen vectors of the matrix $\boldsymbol{A}$ form the new orthogonal dimensions of the transformed process.

superimposed on the scatter diagram of transformed process. Note that the eigen vectors of the matrix form the new orthogonal principal dimensions of the process after transformation.

Example 9.4 Eigen vectors of a Rotation Matrix *R*

As a further example consider a rotation matrix $\boldsymbol{R}$ defined as

$$\boldsymbol{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{9.3}$$

A rotation matrix $\boldsymbol{R}$ can be used to rotate a vector by an angle of θ. The rotation angle can be selected such that at each rotation one vector element is set to zero; this property is used in Givens transformation

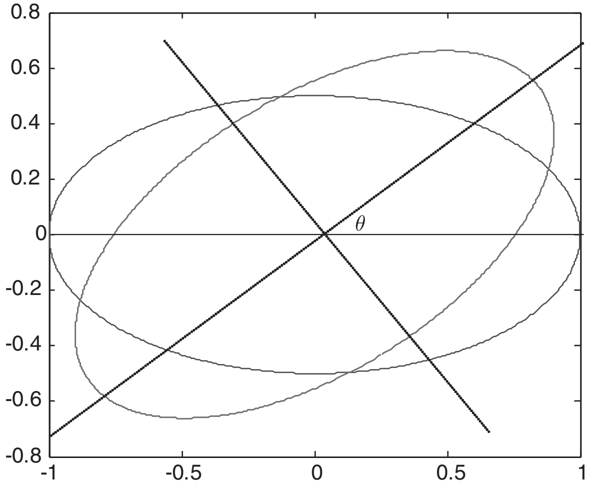


Figure 9.4 An ellipse is rotated by the rotation matrix $\boldsymbol{A} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$.

methods to diagonalise the covariance matrix of a process. Figure 9.4 shows an ellipse rotated anti-clockwise by a rotation matrix with an angle of θ. Note that since the rotation matrix rotates the direction of all real-valued vectors therefore it does not possess a real-valued eigen vector whose direction would be unaffected by the transformation. The eigen vectors of the rotation matrix are complex valued.

9.2 Eigen Vectors and Eigenvalues

Eigen vectors and eigenvalues are used for analysis of square ($N \times N$) matrices such as covariance matrices. For the more general case of $N \times M$ dimensional matrix the singular value decomposition (SVD) method is used.

In general, two sets of eigen vectors are defined for a matrix; the right eigen vectors and the left eigen vectors. The set of right eigenvectors of a matrix are defined as

$$\boldsymbol{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i \qquad i = 1 \ \ldots \ N \tag{9.4}$$

where each eigen vector has a magnitude or norm of 1 (i.e. $\sum_{j=1}^{N} v_{ij}^2 = 1$). Equation (9.4) reveals the main defining characteristic of an eigenvector: when an eigenvector $\boldsymbol{v}_i$ of a matrix $\boldsymbol{A}$ is transformed through the matrix, the direction of the eigen vector is unchanged and only the magnitude of the eigenvector is affected, this explains the name eigen meaning ‘peculiar to’, or ‘characteristic’ or ‘own’ (of a matrix).

As a simple example of an eigen vector, note that the differential operation only affects the magnitude of an exponential, $\frac{d}{dt}e^{\lambda t} = \lambda e^{\lambda t}$, hence the exponential function is an eigen vector of the differential operator.

From Equation (9.4) we can write (by adding the equations for different eigen vectors and values):

$$\boldsymbol{A}\boldsymbol{v}_1 + \cdots + \boldsymbol{A}\boldsymbol{v}_N = \lambda_1 \boldsymbol{v}_1 + \cdots + \lambda_N \boldsymbol{v}_N \tag{9.5}$$

Hence we have

$$\boldsymbol{A}[\boldsymbol{v}_1, \boldsymbol{v}_2 \cdots \boldsymbol{v}_N] = [\boldsymbol{v}_1, \boldsymbol{v}_2 \cdots \boldsymbol{v}_N] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{bmatrix} \tag{9.6}$$

or in a compact matrix notation

$$\boldsymbol{A}\boldsymbol{V}_{\mathrm{R}} = \boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda} \tag{9.7}$$

where the matrix $\boldsymbol{V}_{\mathrm{R}}$ are the set of right eigen vectors and $\boldsymbol{\Lambda}$ is the diagonal eigenvalue matrix. From Equation (9.7) it follows that (by post-multiplying both sides by $\boldsymbol{V}_{\mathrm{R}}^{-1}$)

$$\boldsymbol{A} = \boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{R}}^{-1} \tag{9.8}$$

Similarly, we can define a set of left eigen vectors as

$$\boldsymbol{V}_{\mathrm{L}}\boldsymbol{A} = \boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{L}} \tag{9.9}$$

Pre-multiplying Equation (9.7) by $\boldsymbol{V}_{\mathrm{L}}$ and post multiplying Equation (9.9) by $\boldsymbol{V}_{\mathrm{R}}$ we will have the following Equations

$$\boldsymbol{V}_{\mathrm{L}}\boldsymbol{A}\boldsymbol{V}_{\mathrm{R}} = \boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda} \tag{9.10}$$

$$\boldsymbol{V}_{\mathrm{L}}\boldsymbol{A}\boldsymbol{V}_{\mathrm{R}} = \boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}} \tag{9.11}$$

From Equations (9.10) and (9.11), equating their right hand sides, it follows that

$$\boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}} = \boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda} \tag{9.12}$$

Equation (9.12) is in the form of $\boldsymbol{\Lambda C}=\boldsymbol{C\Lambda}$ where $\boldsymbol{C}=\boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}}$. Hence it follows that since $\boldsymbol{\Lambda}$ is diagonal then for Equation (9.12) to be true in the general case, $\boldsymbol{V}_{\mathrm{L}}\boldsymbol{V}_{\mathrm{R}}$ must also be diagonal identity matrix. It follows that $\boldsymbol{V}_{\mathrm{L}}$ and $\boldsymbol{V}_{\mathrm{R}}$ are orthonormal

$$\boldsymbol{V}_{\mathrm{R}}\boldsymbol{V}_{\mathrm{L}}=\mathrm{I} \tag{9.13}$$

Note from Equation (9.13) that $\boldsymbol{V}_{\mathrm{L}}=\boldsymbol{V}_{\mathrm{R}}^{-1}$ and $\boldsymbol{V}_{\mathrm{R}}=\boldsymbol{V}_{\mathrm{L}}^{-1}$. Post-multiplying both sides of Equation (9.7) by $\boldsymbol{V}_{\mathrm{L}}$ we have

$$\boldsymbol{A}\boldsymbol{V}_{\mathrm{R}}\boldsymbol{V}_{\mathrm{L}}=\boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{L}} \tag{9.14}$$

Using Equation (9.13) in (9.14) we have

$$\boldsymbol{A}=\boldsymbol{V}_{\mathrm{R}}\boldsymbol{\Lambda}\boldsymbol{V}_{\mathrm{L}} \tag{9.15}$$

Note that for a symmetric matrix, such as an autocovariance matrix, $\boldsymbol{V}_{\mathrm{L}}=\boldsymbol{V}_{\mathrm{R}}^{\mathrm{T}}=\boldsymbol{V}^{\mathrm{T}}$. Hence for a symmetric matrix we have

$$\boldsymbol{A}=\boldsymbol{V}\,\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} \tag{9.16}$$

Note that the matrix $\boldsymbol{A}$ can be described in terms of a summation of a set of orthogonal sub-matrices formed from multiplication of the left and right eigenvectors, $\boldsymbol{v}_i\boldsymbol{v}_i{}^{\mathrm{T}}$, as

$$\boldsymbol{A}=\lambda_1\boldsymbol{v}_1\boldsymbol{v}_1{}^{\mathrm{T}}+\lambda_2\boldsymbol{v}_2\boldsymbol{v}_2{}^{\mathrm{T}}+\cdots+\lambda_N\boldsymbol{v}_N\boldsymbol{v}_N{}^{\mathrm{T}} \tag{9.17}$$

where $\left(\boldsymbol{v}_i\boldsymbol{v}_i{}^{\mathrm{T}}\right)\left(\boldsymbol{v}_j\boldsymbol{v}_j{}^{\mathrm{T}}\right)=0$ for $i\neq j$. Note that

$$\boldsymbol{A}\boldsymbol{v}_1\boldsymbol{v}_1{}^{\mathrm{T}}=\lambda_1\boldsymbol{v}_1\boldsymbol{v}_1{}^{\mathrm{T}} \tag{9.18}$$

Hence $\boldsymbol{v}_1\boldsymbol{v}_1{}^{\mathrm{T}}$ may be called as eigen matrix of $\boldsymbol{A}$. Hence in Equation (9.17) the matrix $\boldsymbol{A}$ is expressed in terms of its eigen matrices.

To summarise so far; the determination of the eigenvectors and eigenvalues of a matrix is important in signal processing for matrix diagonalisation. Each eigenvector is paired with a corresponding eigenvalue. Mathematically, two different kinds of eigenvectors need to be distinguished: left eigenvectors and right eigenvectors. However, for many problems it is sufficient to consider only right eigenvectors and their transpose.

Figure 9.5 shows two eigen vectors of a 2 × 2 matrix. Note that as illustrated the eigen vectors have unit magnitude and are orthogonal (perpendicular) to each other.

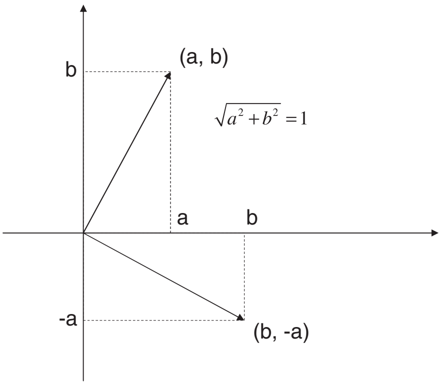


Figure 9.5 A depiction of two orthogonal eigenvectors of a 2 × 2 matrix.

9.2.1 Matrix Spectral Theorem

In Equation (9.17) a matrix $\boldsymbol{A}$ is expressed in terms of a set of orthonormal 'eigen matrices'. The spectral theorem states that a linear transformation of a vector $\boldsymbol{x}$ by a matrix $\boldsymbol{A}$ can be expressed in terms of the sum of a set of orthogonal (subspace) transformations by the eigen matrices as

$$\boldsymbol{Ax} = \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1{}^{\mathrm{T}} \boldsymbol{x} + \lambda_2 \boldsymbol{v}_2 \boldsymbol{v}_2{}^{\mathrm{T}} \boldsymbol{x} + \cdots + \lambda_N \boldsymbol{v}_N \boldsymbol{v}_N{}^{\mathrm{T}} \boldsymbol{x} \tag{9.19}$$

In Equation (9.19) the vector $\boldsymbol{x}$ is decomposed in terms of its projection onto orthogonal subspaces (eigen matrices) of $\boldsymbol{A}$.

Example 9.5 Transforming an input vector x into an eigen vector of a matrix A
Consider a full rank $N \times N$ matrix $\boldsymbol{A}$. The N eigen vectors $\boldsymbol{v}_i$ of this matrix form the N orthogonal dimensions of an N-dimensional space. Hence any N-dimensional vector $\boldsymbol{x}$ can be expressed as a linear combination of the eigen vectors of $\boldsymbol{A}$ as

$$\boldsymbol{x} = b_1 \boldsymbol{v}_1 + b_2 \boldsymbol{v}_2 + \cdots + b_N \boldsymbol{v}_N \tag{9.20}$$

where b_i are the combination weights. Now the transformation of vector $\boldsymbol{x}$ by matrix $\boldsymbol{A}$ can be expressed as

$$\begin{aligned}\boldsymbol{Ax} &= b_1 \boldsymbol{Av}_1 + b_2 \boldsymbol{Av}_2 + \cdots + b_N \boldsymbol{Av}_N \\ &= b_1 \lambda_1 \boldsymbol{v}_1 + b_2 \lambda_2 \boldsymbol{v}_2 + \cdots + b_N \lambda_N \boldsymbol{v}_N \\ &= \lambda_1 \left(b_1 \boldsymbol{v}_1 + \frac{b_2 \lambda_2}{\lambda_1} \boldsymbol{v}_2 + \cdots + \frac{b_N \lambda_N}{\lambda_1} \boldsymbol{v}_N \right)\end{aligned} \tag{9.21}$$

Passing the vector $\boldsymbol{x}$ through the matrix $\boldsymbol{A}$ for a large number of n times results in

$$\boldsymbol{A}^n \boldsymbol{x} = \lambda_1^n \left(b_1 \boldsymbol{v}_1 + b_2 \frac{\lambda_2^n}{\lambda_1^n} \boldsymbol{v}_2 + \cdots + b_N \frac{\lambda_N^n}{\lambda_1^n} \boldsymbol{v}_N \right) \approx \lambda_1^n b_1 \boldsymbol{v}_1 \tag{9.22}$$

where it is assumed that λ_1 is the largest eigenvalue. Note that as the vector $\boldsymbol{x}$ is passed through the matrix $\boldsymbol{A}$ for a large number of times it turns into the eigen vector of $\boldsymbol{A}$ associated with the largest eigenvalue. Note that the approximation in Equation (9.21) does not hold if the signal has two or more eigenvalues with equal magnitudes.

9.2.2 Computation of Eigenvalues and Eigen Vectors

An eigenvalue λ_i of a matrix $\boldsymbol{A}$ can be obtained from the characteristic polynomial of $\boldsymbol{A}$ as explained in the followings. Rearranging Equation (9.4) we obtain

$$(\boldsymbol{A} - \lambda_i \mathbf{I}) \boldsymbol{v}_i = 0 \qquad i = 1, \ldots, N \tag{9.23}$$

Hence we have

$$\det(\boldsymbol{A} - \lambda_i \mathbf{I}) = 0 \qquad i = 1, \ldots, N \tag{9.24}$$

The determinant, $\det(\cdot)$ of the matrix equation gives the characteristic polynomial in terms of the powers of the eigenvalues. For example consider the following small 2 × 2 matrix

$$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \tag{9.25}$$

An eigenvalue of this matrix is obtained from

$$\det\begin{pmatrix} a-\lambda & b \\ c & d-\lambda \end{pmatrix} = 0 \tag{9.26}$$

or

$$(a-\lambda)(d-\lambda) - bc = 0 \tag{9.27}$$

The characteristic polynomial equation of the eigen matrix is given by

$$\lambda^2 - (a+d)\lambda + (ad-bc) = 0 \tag{9.28}$$

For large matrices the eigenvalues are obtained using an iterative root-finding numerical solutions.

Once the eigenvalues of the matrix are obtained, the corresponding eigen vectors can be obtained by solving the following equation set of linear equations

$$(\boldsymbol{A} - \lambda_i \mathbf{I})\boldsymbol{v}_i = 0 \qquad i = 1, \ldots, N \tag{9.29}$$

where for each eigenvalue of λ_i we have a set of linear equations with $\boldsymbol{v}_i$ as the vector that contains the unknown values to be determined.

9.3 Principal Component Analysis (PCA)

Principal component analysis (PCA), Figure 9.6, is a popular data analysis method used for such purposes as signal compression, feature extraction, model order estimation and signal and noise separation. PCA is also called the **Karhunen-Loève transform** (or KLT, named after Kari Karhunen and Michael Loève) or the **Hotelling transform** (after Harold Hotelling).

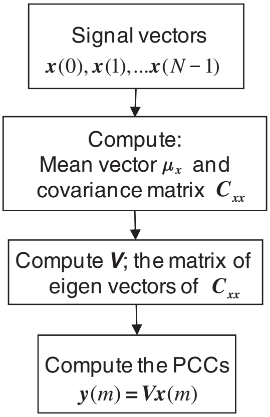

Signal vectors
$\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots \boldsymbol{x}(N-1)$
Compute:
Mean vector μ_x and
covariance matrix $\boldsymbol{C}_{xx}$
Compute $\boldsymbol{V}$; the matrix of
eigen vectors of $\boldsymbol{C}_{xx}$
Compute the PCCs
$\boldsymbol{y}(m) = \boldsymbol{V}\boldsymbol{x}(m)$

Figure 9.6 A block diagram illustration of PCA process, PCC = principal component coefficient.

The principal components (PCs) of a signal process are a set of orthogonal components obtained from an eigen analysis of its covariance matrix. The purpose of PCA is to transform a signal, such as speech or image, and represent it in terms of an orthogonal set of principal components.

The PCA process will reveal the significant uncorrelated structures of the signal process; in the sense that the most important (principal) components of the signal will be along the dimensions with the largest covariance values. Furthermore, as the principal components (PC) dimensions are orthogonal, the signal along each dimension can be processed independently. The PCA achieves the following:

(1) PCA transforms a process such that the data are represented along a new set of orthogonal dimensions with a diagonal covariance matrix.
(2) The PC coefficient with the largest variance is the first principal component, the PC coefficient with the second largest variance is the second most important PC and so on.
(3) The transformed PCs are uncorrelated and can be processed independently.
(4) For an over-modelled process, i.e. where the dimension of the correlation matrix (M) is bigger than the actual number of significant components of the signal (N), the last $N - M$ principal components represent the noise variances.

9.3.1 Computation of PCA

Assume we have N samples of a vector process $\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(N-1)$. These vectors may be speech frames or sub blocks of an image. The first step in PCA analysis is to obtain an estimate of the mean of the signal vector as

$$\boldsymbol{\mu}_x = \frac{1}{N}\sum_{m=0}^{N-1} \boldsymbol{x}(m) \tag{9.30}$$

An estimate of the covariance matrix of the signal vector is then obtained as

$$\boldsymbol{C}_{xx} = \frac{1}{N}\sum_{m=0}^{N-1} \left(\boldsymbol{x}(m) - \boldsymbol{\mu}_x\right)\left(\boldsymbol{x}(m) - \boldsymbol{\mu}_x\right)^T \tag{9.31}$$

The next step in PCA is an eigen analysis of the covariance matrix $\boldsymbol{C}_{xx}$. The covariance matrix is expressed in terms of its eigen vectors and eigenvalues as

$$\boldsymbol{C}_{xx} = \boldsymbol{V}\,\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{V} \tag{9.32}$$

Note that since the covariance matrix is real and symmetric its eigen vectors are real and orthonormal. From Equation (9.32) it can be shown that the transformation of the vector process $\boldsymbol{x}$ by the matrix of eigen vectors $\boldsymbol{V}$ diagonalises its covariance matrix as

$$\boldsymbol{y} = \boldsymbol{V}\boldsymbol{x} \tag{9.33}$$

$$\boldsymbol{C}_{yy} = E(\boldsymbol{V}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}) = \boldsymbol{V}\boldsymbol{C}_{xx}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\ \boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{\Lambda} \tag{9.34}$$

where the operator $E(\cdot)$ represents the expectation operation and $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \mathbf{I}$.

If we wish to 'sphere a process', i.e. to diagonalise and also normalise the covariance matrix of the process, then we need to transform the signal as

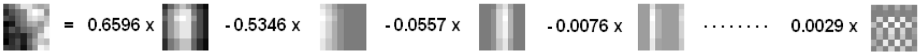
$$\boldsymbol{y} = \boldsymbol{C}_{xx}^{-1/2}\boldsymbol{x} = (\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V})\boldsymbol{x} \tag{9.35}$$

$$\begin{aligned}\boldsymbol{C}_{yy} &= E(\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}) = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}\boldsymbol{C}_{xx}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\\ &= \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{-1/2} = \mathbf{I}\end{aligned} \tag{9.36}$$

As mentioned, the process of diagonalising and equalising the covariance matrix of a process is known as sphereing.

9.3.2 PCA Analysis of Images: Eigen-Image Representation

PCA analysis can be used to decompose an image into a set of orthogonal principal component images also known as eigen images. Eigen images can be used for image coding, image denoising, or as features for image classification. For example an image subblock can be reconstructed as a weighted function of the principal eigen images as



To obtain the set of eigen-images $[\boldsymbol{E}_{ij}]$ for a given image $\boldsymbol{A}$ of size $r_0 \times c_0$ pixels, the image is first divided into N sub-images (subblocks) $\boldsymbol{A}_k$ of size $r \times c$ (typically 8×8 or 16×16). The mean of the sub-images is obtained as

$$\boldsymbol{A}_{mean} = \frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{A}_k \tag{9.37}$$

The mean image is then removed from each sub-image as

$$\bar{\boldsymbol{A}}_k = \boldsymbol{A}_k - \boldsymbol{A}_{mean} \tag{9.38}$$

An $r \times r$ covariance matrix for the rows of the sub-images is obtained

$$\boldsymbol{C}_r = \frac{1}{N}\sum_{k=0}^{N-1}\bar{\boldsymbol{A}}_k\bar{\boldsymbol{A}}_k^{\mathrm{T}} \tag{9.39}$$

Similarly, a $c \times c$ covariance matrix for the columns of the sub-images is obtained

$$\boldsymbol{C}_c = \frac{1}{N}\sum_{k=0}^{N-1}\bar{\boldsymbol{A}}_k^{\mathrm{T}}\bar{\boldsymbol{A}}_k \tag{9.40}$$

The row and column covariance matrices are then subjected to eigen analysis to yield r row eigen vectors $[\boldsymbol{r}_i, 1 \le i \le r]$ and c column eigen vectors $[\mathbf{c}_j, 1 \le j \le c]$.

A total of $r \times c$ eigen image $\boldsymbol{E}_{ij}$ is defined as the product of each row eigen vector with each column eigen vector as

$$\boldsymbol{E}_{ij} = \boldsymbol{r}_i\boldsymbol{c}_j^T \qquad 1 \le i \le r \quad 1 \le j \le c \tag{9.41}$$

Figure 9.7 shows an example of eigen images for an image of the brain. Figure 9.8 shows the original brain scan image and the image reconstructed from the principal eigen images.

9.3.3 PCA Analysis of Speech in White Noise

PCA analysis can be used to decompose a noisy speech signal; retain the most significant signal components that are ‘above’ a desired signal to noise ratio threshold and then discard the remaining noisy components.

In this method a PCA of the correlation matrix of the noisy speech signal is performed. First speech is segments into frames of N samples. For each frame a correlation matrix for the signal is obtained. The correlation matrix is then decomposed into a matrix of eigen vectors and their corresponding eigenvalues. The most significant components of the signal, corresponding to largest eigenvalues of the autocorrelation matrix, are retained whereas the noisy components corresponding to the smallest eigenvalues are discarded. The signal is then expressed as a combination of its most significant PC eigen vectors where the combination factors are obtained as normalised cross correlation of the signal and each PC eigen vector.

Note that for white noise assuming that the dimensions of the correlation matrix is bigger that the dimensions of significant correlation lags in the signal (say < 14) then the smallest eigenvalues correspond to the variance of noise.

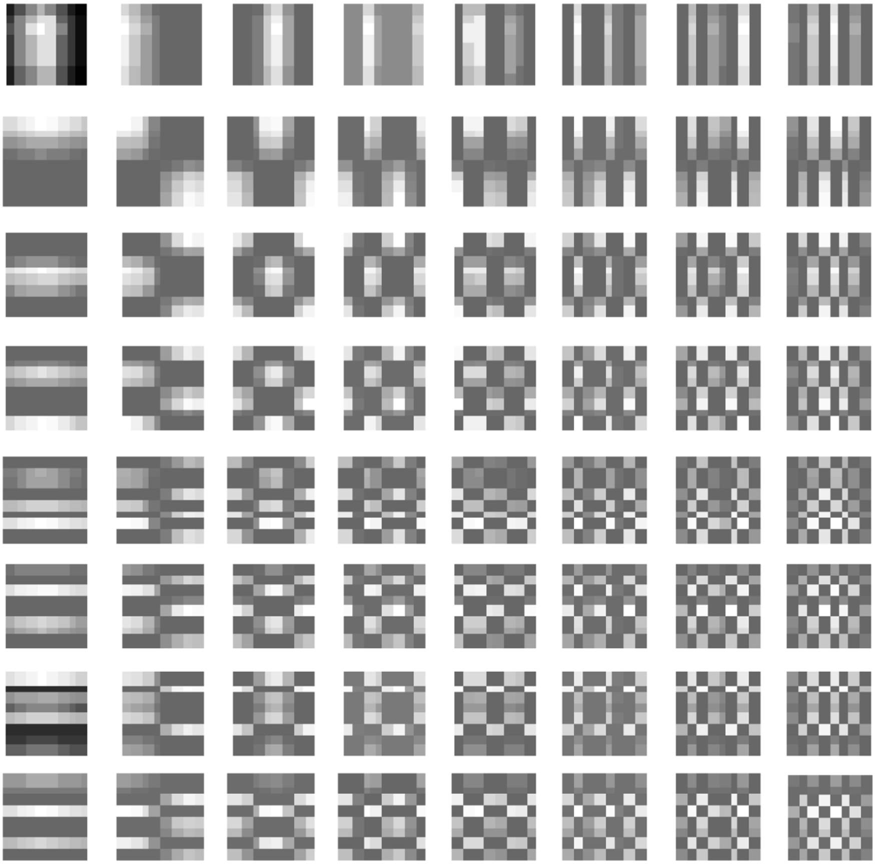


Figure 9.7 64 eigen images of size 8 by 8 in the descending order of associated eigenvalues.

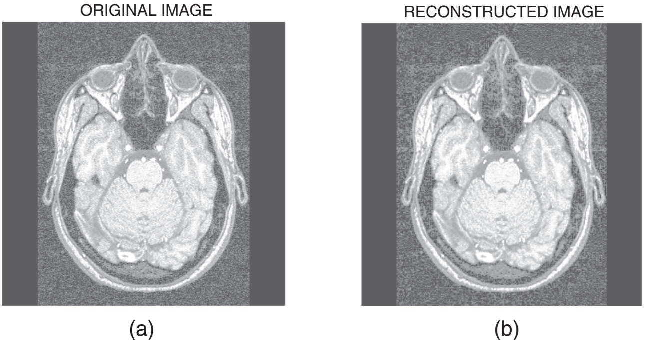


Figure 9.8 (a) An original image scan of brain, (b) reconstructed from the 4 most significant eigen-images associated with subblocks of 8 by 8. Note that images blocks of size 8 by 8 block have 64 eigen images. Reproduced by permission of © 2008 Oldrich Vysata M.D. Neurocenter Caregroup, Rychnov nad Kneznou, Prague.

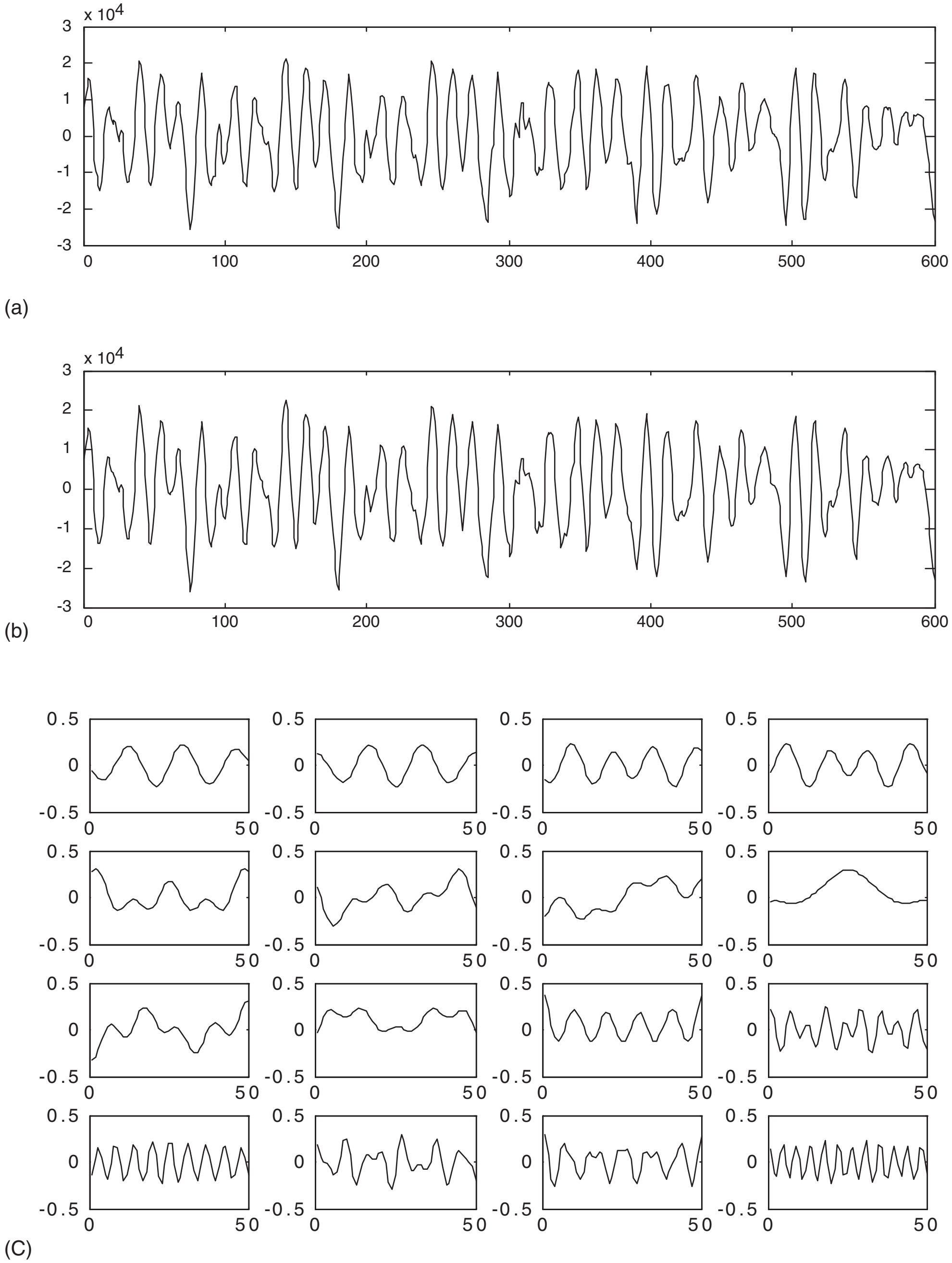


Figure 9.9 Principal component analysis of time domain speech signals. (a) original speech was segmented into segments of 50 samples long and its covariance matrix was obtained for eigen analysis, (b) shows the speech reconstructed from the 16 principal components eigen vectors, (c) shows 16 principal eigen vectors of speech.

Figure 9.9 shows the use of PCA in reconstructing speech from its most significant components. In this case, speech was segmented into segments of 50 samples long and correlation matrices of size 50 × 50 were calculated as average values over segments of 400 speech samples. Then segments of 50 samples long of speech were expressed in terms of the 16 most significant eigen vectors. Note that the choices of segment size can be varied so long as it conforms to the stationary assumption of PCA.

Figure 9.10.a shows a speech signal contaminated with additive white noise. Figure 9.10.b shows the same signal reconstructed from 16 principal components after a PCA was applied to a 50 × 50 covariance matrix of the speech signal. Note that the noise has been significantly reduced by retaining the PCs with relatively high SNRs and discarding the PCs with lowers SNRs.

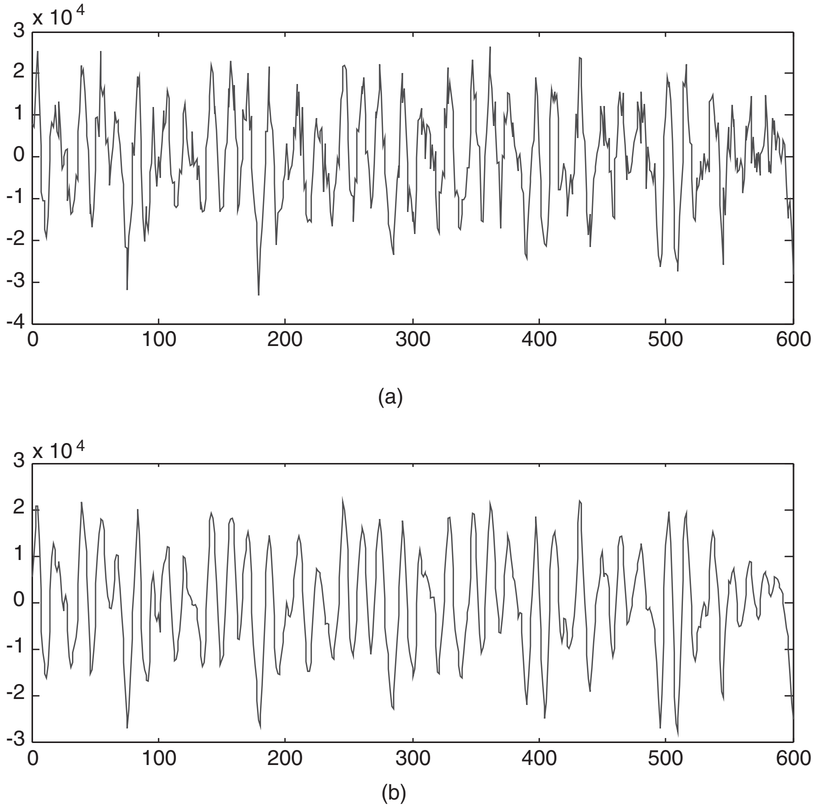


Figure 9.10 Illustration of the use of PCA in removing white noise. (a) noisy speech, (b) segments of 50 sample-long of speech reconstructed from 16 most significant principal components.

9.4 Summary

The in-depth understanding and utilisation of the linear system theory is of immense importance in the study and application of digital signal processing. A central concept in linear system theory is the concept of eigen analysis. Eigen analysis provides the tool for decomposition of a signal in terms of a set of orthogonal basis functions. Eigen vector analysis can be used to white or sphere a signal process, sphereing a vector-valued signal is the process of whitening the signal in such a way that all its components have unit variance. Eigen vector analysis is also used in principal component analysis (PCA) to determine the most significant orthogonal basis components of a signal process. Independent component analysis can be viewed as an extension of eigen analysis or PCA, however, there are substantial differences. Unlike PCA, ICA can be used to whiten signals that are non-Gaussian. One important application of ICA is in blind source separation (BSS), another application is in principle feature extraction for non-Gaussian signals (such as eigen image extraction) as a more efficient alternative to PCA. The main feature of ICA is that it employs non-linear influence/contrast functions in the process of estimation of the optimal linear whitening transformation.

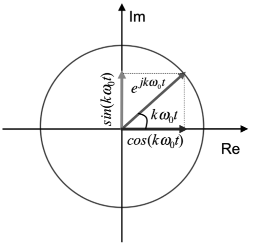
Bibliography

Golub, G.H., Van Loan, C.F. (1996) *Matrix Computations*. 3rd edn, Johns Hopkins University Press.

Jolliffe, I.T. (2002) *Principal Component Analysis*. Springer-Verlag.

Marcus, M. and Minc, H. (1988) *Introduction to Linear Algebra*. New York: Dover, p. 145.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1992) ‘Eigensystems’ Chapter 11 in *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd edn, Cambridge, England: Cambridge University Press, pp. 449–489.



10

Power Spectrum Analysis

The power spectrum reveals the existence, or the absence, of repetitive patterns, vibrations 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 model-based 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 eigen-analysis, for the estimation of sinusoids observed in additive white noise.

10.1 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: correlation is a function of time domain and power spectrum is a function of frequency domain. Correlation and 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 power spectrum, 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 in its ability to reveal spectral structures that may be used to characterise a signal. This is illustrated in Figure 10.1 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 10.1(a) the signal

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

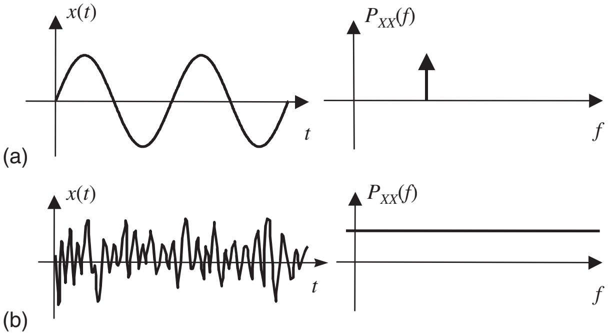


Figure 10.1 The concentration/spread of power in frequency indicates the correlated or random character of a signal: (a) a predictable signal $x(t)$ and its power spectrum $P_{XX}(f)$, (b) a random signal and its power spectrum.

power is concentrated in one frequency. For a purely random signal as shown in Figure 10.1(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, the more concentrated its power spectrum, and conversely the more random or unpredictable a signal, the more spread 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.

10.2 Fourier Series: Representation of Periodic Signals

A periodic signal can be described in terms of a series of harmonically related (i.e. integer multiples of a fundamental frequency) sine and cosine waves.

The following three sinusoidal functions form the *basis functions* for Fourier analysis.

$$x_1(t) = \cos \omega_0 t \tag{10.1}$$

$$x_2(t) = \sin \omega_0 t \tag{10.2}$$

$$x_3(t) = \cos \omega_0 t + j \sin \omega_0 t = e^{j\omega_0 t} \tag{10.3}$$

A cosine function is an even function with respect to the vertical axis (amplitude) at time $t = 0$ and a sine function is an odd function. A weighted combination of a sine and a cosine at angular frequency ω_0 can model any phase of a sinusoidal signal component of $x(t)$ at that frequency.

Figure 10.2(a) shows the sine and the cosine components of the complex exponential (cisoidal) signal of Equation (10.3), and Figure 10.2(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 a period of $T_0 = 1/F_0$ and an angular frequency of $\omega_0 = 2\pi F_0$ radians/second, where F_0 is the frequency in Hz.

10.2.1 The Properties of Fourier's Sinusoidal Basis Functions

The following properties make the sinusoids an ideal choice as the elementary building block basis functions for signal analysis and synthesis.

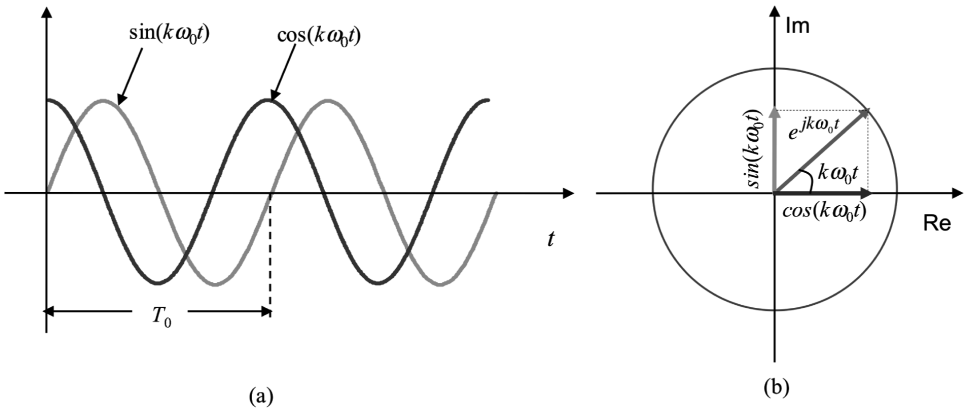


Figure 10.2 Fourier basis functions: (a) real and imaginary components of a complex sinusoid, (b) vector representation of a complex exponential. If the cosine is considered as the in-phase component then the sine is the quadrature component.

(1) Orthogonality; two sinusoidal functions of *different* frequencies have the following orthogonal property:

$$\int_{-\infty}^{\infty} \sin(\omega_1 t)\sin(\omega_2 t)dt = -\frac{1}{2}\int_{-\infty}^{\infty} \cos(\omega_1+\omega_2)t\,dt + \frac{1}{2}\int_{-\infty}^{\infty} \cos(\omega_1-\omega_2)t\,dt = 0 \tag{10.4}$$

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 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 audio 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 $e^{j\omega t}$ 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 the decomposition of a signal in terms of orthogonal cosine (in-phase) and sine (quadrature) components.

(3) Sinusoidal functions are infinitely differentiable. This is a useful property, as most signal analysis and synthesis methods require the signals to be differentiable.

(4) A useful consequence of transforms, such as the Fourier and Laplace transforms, is that relatively difficult differential analysis on the time domain signal become relatively simple algebraic operations on the transformed signal.

10.2.2 The Basis Functions of Fourier Series

Associated with the complex exponential function $e^{j\omega_0 t}$ is a set of periodic and harmonically related complex exponentials of the form $[1, e^{\pm j\omega_0 t}, e^{\pm j2\omega_0 t}, e^{\pm j3\omega_0 t}, \ldots]$ 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 series analysis. Any linear combination of these signals, of the form $x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$, is also a periodic signal with a period of T_0. Conversely any periodic signal $x(t)$ can be synthesised from a linear combination of harmonically related exponentials.

The Fourier series representation of a periodic signal, with a period of T_0 and angular frequency $\omega_0 = 2\pi/T_0 = 2\pi F_0$, is given by the following synthesis and analysis equations:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \qquad \textbf{Synthesis equation} \tag{10.5}$$

$$c_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-jk\omega_0 t} dt \quad k = \cdots, -1, 0, 1, \cdots \qquad \textbf{Analysis equation} \tag{10.6}$$

10.2.3 Fourier Series Coefficients

The complex-valued Fourier series coefficient c_k conveys the amplitude (a measure of the strength) and the phase (or time delay) of the frequency content of the signal at frequency $k\omega_0$ Hz. Note from the Fourier analysis Equation (10.6), 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 (10.5) as the sum of its constituent harmonics is referred to as the *complex Fourier series* representation. The set of complex coefficients $\ldots, c_{-1}, c_0, c_1, \ldots$ is known as the *frequency spectrum* of the signal.

Equation (10.5) can be used as a synthesizer (as in a music synthesizer) to generate a signal as a weighted combination of its elementary frequencies. Note from Equations (10.5) and (10.6) that the complex exponentials that form a periodic signal occur only at discrete frequencies which are integer multiple harmonics of the fundamental frequency ω_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 $\omega_0 = 2\pi/T_0$.

Note from Equations (10.5) and (10.6) that the complex exponentials that form a periodic signal occur only at discrete frequencies which are integer multiple harmonics of the fundamental frequency ω_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 $\omega_0 = 2\pi/T_0$.

10.3 Fourier Transform: Representation of Non-periodic Signals

The Fourier representation of non-periodic signals can be obtained by considering a non-periodic 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 hence it is non-periodic.

The Fourier series representation of periodic signals consist of harmonically related sinusoidal signals with a discrete spectra, where the spectral lines are spaced at integer multiples of the fundamental frequency F_0. Now consider the discrete spectra of a periodic signal with a period of T_0, as shown in Figure 10.3(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 non-periodic) the discrete spectral lines merge and form a continuous spectrum as shown in Figure 10.3(b).

Therefore, the Fourier equations for a non-periodic signal (known as the Fourier transform), must reflect the fact that the frequency spectrum of a non-periodic signal is a continuous function of frequency. Hence, to obtain the Fourier transform relations the discrete-frequency variables and the discrete summation operations in the Fourier series Equations (10.5) and (10.6) 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 harmonics 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 $X(f)$.

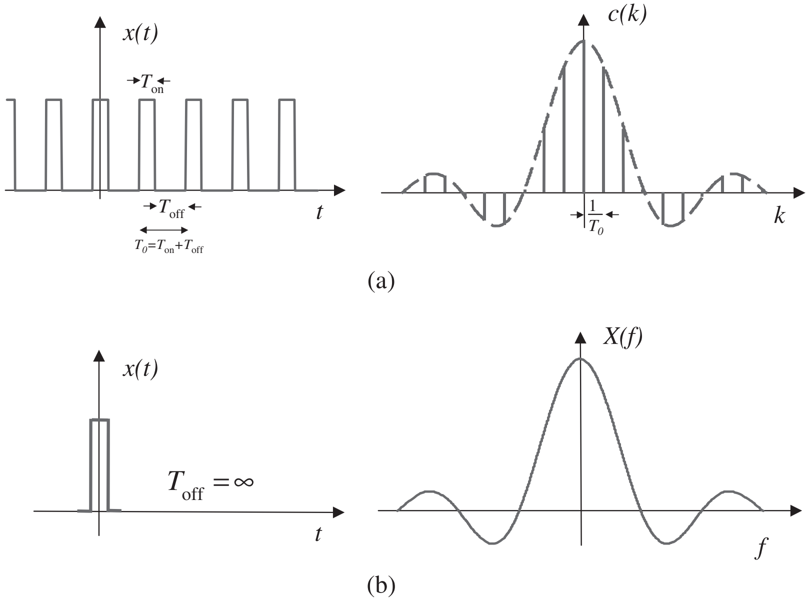


Figure 10.3 (a) A periodic pulse train and its line spectrum. (b) A single pulse from the periodic train in (a) with an imagined 'off' duration of infinity; its spectrum is the envelope of the spectrum of the periodic signal in (a).

The Fourier analysis and synthesis equations for non-periodic signals, known as the *Fourier transform pair*, are given by

Fourier Transform (Analysis) Equation

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \tag{10.7}$$

Inverse Fourier Transform (Synthesis) Equation

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df \tag{10.8}$$

Note from Equation (10.7), 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

$$Signal\ Energy = \int_{-\infty}^{\infty} |x(t)|^2\, dt < \infty \tag{10.9}$$

10.3.1 Discrete Fourier Transform

Discrete Fourier transform (DFT) is the Fourier transform adapted for digital signal processing. The DFT of a continuous-time signal can be derived by three operations of: (1) sampling in time, (2) segmenting the sampled signal into segments of length N samples and (3) sampling the spectrum of the discrete-time signal segment in frequency domain.

Note that just as sampling a signal in time (or space) renders its spectrum a *periodic* function of frequency, it follows that sampling a signal in frequency renders it periodic in time (or space). This is also an aspect of the principle of duality.

The first two steps yields the Fourier transform of a sampled and windowed signal as

$$X(f) = \int_{-\infty}^{\infty} x(t) \underbrace{\sum_{m=0}^{N-1} \delta(t - mT_s)}_{N-\text{point sampling}} e^{-j2\pi ft} dt = \sum_{m=0}^{N-1} x(mT_s) e^{-j2\pi fmT_s} \tag{10.10}$$

where $\delta(.)$ is the Kronecker delta function. Without loss of generality it is usually assumed that the sampling period T_s and hence the sampling frequency F_s is equal to 1, i.e. $T_s = 1/F_s = 1$.

Note that in general $X(f)$ is a continuous and periodic function of the frequency variable f. The periodicity in $X(f)$ is introduced as a result of sampling. The final step is sampling in frequency domain that is evaluating $X(f)$ at discrete frequencies $f = kF_s/N$ where k is an integer and F_s is the sampling rate

$$X(k) = X(f)\delta(f - \frac{k}{N}F_s) \tag{10.11}$$

Note that $X(k)$ is shorthand for $X(kFs/N)$, that is the k^{th} discrete frequency corresponds to an actual frequency of kFs/N.

As illustrated in Figure 10.4 the input to DFT is N samples of a discrete signal $[x(0), \ldots, x(N-1)]$ and the output consists of N uniformly-spaced samples $[X(0), \ldots, X(N-1)]$ of the frequency spectrum of the input. When a non-periodic continuous-time signal is sampled, its Fourier transform becomes a periodic but *continuous* function of frequency. As shown above the discrete Fourier transform is derived from sampling the Fourier transform of a discrete-time signal at N discrete-frequencies corresponding to integer multiples of the frequency sampling interval $2\pi/N$. *The DFT is effectively the Fourier series of a sampled signal.*

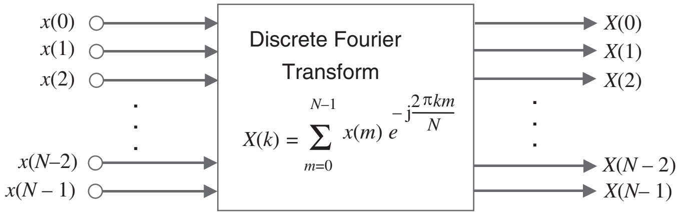

x(0)
x(1)
x(2)
x(N–2)
x(N – 1)
Discrete Fourier Transform
$X(k) = \sum_{m=0}^{N-1} x(m)\, e^{-j\frac{2\pi km}{N}}$
X(0)
X(1)
X(2)
X(N – 2)
X(N– 1)

Figure 10.4 Illustration of the DFT as a parallel-input parallel-output signal processor.

For a finite duration discrete-time signal $x(m)$ of length N samples, the discrete Fourier transform (DFT) and its inverse (IDFT) are defined as

Discrete Fourier transform (DFT) analysis equation

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j\frac{2\pi}{N}mk} \qquad k = 0, \ldots, N\text{–}1 \tag{10.12}$$

Inverse discrete Fourier transform (IDFT) synthesis equation

$$x(m) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}mk} \quad m = 0, \ldots, N\text{–}1 \tag{10.13}$$

Note that the basis functions of a DFT are: 1, $e^{-j\frac{2\pi}{N}}, e^{-j\frac{4\pi}{N}}, \ldots, e^{-j\frac{2(N-1)\pi}{N}}$.

The DFT equation can be written in the form of a linear system transformation, $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{x}$, as the transformation of an input vector $\boldsymbol{x} = [x(0)\ x(1)\ \ldots\ x(N-1)]$ to an output vector $\boldsymbol{X} = [X(0)\ X(1)\ \ldots\ X(N-1)]$ as

$$\underbrace{\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-2) \\ X(N-1) \end{bmatrix}}_{\text{Output Vector, } X} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & \cdots & e^{-j\frac{2(N-2)\pi}{N}} & e^{-j\frac{2(N-1)\pi}{N}} \\ 1 & e^{-j\frac{4\pi}{N}} & e^{-j\frac{8\pi}{N}} & \cdots & e^{-j\frac{4(N-2)\pi}{N}} & e^{-j\frac{4(N-2)\pi}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & e^{-j\frac{2(N-2)\pi}{N}} & e^{-j\frac{4(N-2)\pi}{N}} & \cdots & e^{-j\frac{2(N-2)^2\pi}{N}} & e^{-j\frac{2(N-2)(N-1)\pi}{N}} \\ 1 & e^{-j\frac{2(N-1)\pi}{N}} & e^{-j\frac{4(N-1)\pi}{N}} & \cdots & e^{-j\frac{2(N-1)(N-2)\pi}{N}} & e^{-j\frac{(N-1)^2\pi}{N}} \end{bmatrix}}_{\textit{Fourier Transform Matrix, W}} \underbrace{\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-2) \\ x(N-1) \end{bmatrix}}_{\text{Input Vector, } x} \tag{10.14}$$

where the elements of the transformation matrix are $w_{km} = e^{-j\frac{2\pi}{N}km}$ where m is the discrete-time index and k is the discrete-frequency index.

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* $[x(0), \ldots, x(N-1)]$ *is periodic with a period equal to the observation window length of N samples.*

10.3.2 Frequency-Time Resolutions: The Uncertainty Principle

An important practical issue in applications of DFT is the concept of resolution, that is frequency versus time (or space) resolutions. Resolution is the ability to resolve details of a signal in time or frequency or space. For example a satellite imaging system may have a resolution of 1 metre which means that it cannot capture details smaller than 1 metre. Similarly in this section we will show that the frequency resolution of DFT is the width of each DFT bin between two successive discrete frequencies. The ability to resolve two closely spaced frequencies is inversely proportional to the DFT length, that is the length of the input signal.

The DFT length and hence the resolution of DFT is limited by the non-stationary character of most signals and also by the allowable delay in communication systems. Signals such as speech, music or image are generated by non-stationary – i.e. time-varying and/or space varying – systems or processes. For example speech is composed of a sequence of short-duration sounds called phonemes each of which has a different spectral-temporal composition; music is composed of notes and sounds whose frequency compositions vary with time and an image is composed of various objects. As Fourier transform assumes that the signals within the DFT window of N samples are stationarity, the spectra of non-stationary signal events would be averaged over the duration of window and would not be resolved and shown as separate or distinct signal events.

When using the DFT it is desirable to have a high resolution in time or space (that means small window size or number of input samples N) in order to obtain the details of the spectral characteristics of each individual elementary event, sound or object in the input signal. However there is a fundamental trade-off between the length, of the input signal (i.e. the time or space resolution) and the frequency resolution of the DFT output.

The DFT takes as the input a window of N uniformly spaced discrete-time samples $[x(0), x(1), \ldots, x(N-1)]$ with a total duration of $\Delta T = N.T_s$, and outputs N spectral samples $[X(0), X(1), \ldots, X(N-1)]$ 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

$$\Delta f = \frac{F_s}{N} = \frac{1}{NT_s} = \frac{1}{\Delta T} \tag{10.15}$$

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.

10.3.3 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

$$E(f) = |X(f)|^2 = \left| \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi fm} \right|^2 \tag{10.16}$$

The energy spectrum of $x(m)$ may be expressed as the Fourier transform of the autocorrelation function of $x(m)$:

$$\begin{aligned} E(f) &= |X(f)|^2 = X(f)X^*(f) \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \end{aligned} \tag{10.17}$$

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:

$$P_{XX}(f) = \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \tag{10.18}$$

where the autocorrelation function $r_{xx}(m)$ is defined as

$$r_{xx}(m) = \mathcal{E}[x(m)x(m+k)] \tag{10.19}$$

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

$$\hat{r}_{xx}(m) = \frac{1}{N-|m|} \sum_{k=0}^{N-|m|-1} x(k)x(k+m), \quad k = 0, \ldots, N-1 \tag{10.20}$$

Equation (10.20), 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

$$w(m) = \begin{cases} 1 - \frac{|m|}{N}, & |m| \leq N-1 \\ 0, & \text{otherwise} \end{cases} \tag{10.21}$$

Multiplication of Equation (10.20) by the window of Equation (10.21) yields

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-|m|-1} x(k)x(k+m) \tag{10.22}$$

The expectation of the windowed correlation estimate $\hat{r}_{xx}(m)$ is given by

$$\begin{aligned} \mathcal{E}\left[\hat{r}_{xx}(m)\right] &= \frac{1}{N} \sum_{k=0}^{N-|m|-1} \mathcal{E}[x(k)x(k+m)] \\ &= \left(1 - \frac{|m|}{N}\right) r_{xx}(m) \end{aligned} \tag{10.23}$$

In Jenkins and Watts, it is shown that the variance of $\hat{r}_{xx}(m)$ is given by

$$\text{Var}\left[\hat{r}_{xx}(m)\right] \approx \frac{1}{N} \sum_{k=-\infty}^{\infty} \left[r_{xx}^2(k) + r_{xx}(k-m)r_{xx}(k+m)\right] \tag{10.24}$$

From Equations (10.23) and (10.24), $\hat{r}_{xx}(m)$ is an asymptotically unbiased and consistent estimate.

10.4 Non-Parametric 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 1891. The periodogram is defined as

$$\begin{aligned} \hat{P}_{XX}(f) &= \frac{1}{N} \left| \sum_{m=0}^{N-1} x(m) e^{-j2\pi fm} \right|^2 \\ &= \frac{1}{N} |X(f)|^2 \end{aligned} \tag{10.25}$$

The power-spectral density function, or power spectrum for short, defined in Equation (10.25), is the basis of non-parametric 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.

10.4.1 The Mean and Variance of Periodograms

The mean of the periodogram is obtained by taking the expectation of Equation (10.25):

$$\begin{aligned} \mathcal{E}[\hat{P}_{XX}(f)] &= \frac{1}{N} \mathcal{E}\left[|X(f)|^2\right] \\ &= \frac{1}{N} \mathcal{E}\left[\sum_{m=0}^{N-1} x(m) e^{-j2\pi fm} \sum_{n=0}^{N-1} x(n) e^{j2\pi fn}\right] \\ &= \sum_{m=-(N-1)}^{N-1} \left(1 - \frac{|m|}{N}\right) r_{xx}(m) e^{-j2\pi fm} \end{aligned} \tag{10.26}$$

As the number of signal samples N increases, we have

$$\lim_{N\to\infty} \mathcal{E}[\hat{P}_{XX}(f)] = \sum_{m=-\infty}^{\infty} r_{xx}(m)e^{-j2\pi fm} = P_{XX}(f) \tag{10.27}$$

For a Gaussian random sequence, the variance of the periodogram can be obtained as

$$\mathrm{Var}[\hat{P}_{XX}(f)] = P_{XX}^2(f)\left[1+\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{10.28}$$

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, are 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.

10.4.2 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

$$\hat{P}_{XX}^B(f) = \frac{1}{K}\sum_{i=1}^{K}\hat{P}_{XX}^{(i)}(f) \tag{10.29}$$

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

$$\begin{aligned}\mathcal{E}[\hat{P}_{XX}^B(f)] &= \mathcal{E}[\hat{P}_{XX}^{(i)}(f)]\\ &= \sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)r_{xx}(m)e^{-j2\pi fm}\\ &= \frac{1}{N}\int_{-1/2}^{1/2} P_{XX}(\nu)\left[\frac{\sin\pi(f-\nu)N}{\sin\pi(f-\nu)}\right]^2 d\nu\end{aligned} \tag{10.30}$$

where $(\sin\pi fN/\sin\pi f)^2/N$ is the frequency response of the triangular window $1-|m|/N$. From Equation (10.30), 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

$$\mathrm{Var}\left[\hat{P}_{XX}^B(f)\right] = \frac{1}{K}P_{XX}^2(f)\left[1+\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{10.31}$$

10.4.3 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

$$x_i(m) = x(m+iD), \quad m=0,\ldots,N-1, \quad i=0,\ldots,K-1 \tag{10.32}$$

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

$$\hat{P}_{XX}^{(i)}(f) = \frac{1}{NU}\left|\sum_{m=0}^{N-1} w(m)x_i(m)e^{-j2\pi fm}\right|^2 \tag{10.33}$$

where $w(m)$ is the window function and U is the power in the window function, given by

$$U = \frac{1}{N}\sum_{m=0}^{N-1} w^2(m) \tag{10.34}$$

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:

$$\hat{P}_{XX}^{W}(f) = \frac{1}{K}\sum_{i=0}^{K-1} \hat{P}_{XX}^{(i)}(f) \tag{10.35}$$

Using Equations (10.33) and (10.35), the expectation of $\hat{P}_{XX}^{W}(f)$ can be obtained as

$$\begin{aligned}
\mathcal{E}[P_{XX}^{W}(f)] &= \mathcal{E}[\hat{P}_{XX}^{(i)}(f)] \\
&= \frac{1}{NU}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w(n)w(m)\mathcal{E}[x_i(m)x_i(n)]e^{-j2\pi f(n-m)} \\
&= \frac{1}{NU}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} w(n)w(m)r_{xx}(n-m)e^{-j2\pi f(n-m)} \\
&= \int_{-1/2}^{1/2} P_{XX}(v)W(v-f)dv
\end{aligned} \tag{10.36}$$

where

$$W(f) = \frac{1}{NU}\left|\sum_{m=0}^{N-1} w(m)e^{-j2\pi fm}\right|^2 \tag{10.37}$$

and the variance of the Welch estimate is given by

$$\mathrm{Var}[\hat{P}_{XX}^{W}(f)] = \frac{1}{K^2}\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} \mathcal{E}\left[\hat{P}_{XX}^{(i)}(f)\hat{P}_{XX}^{(j)}(f)\right] - \left(\mathcal{E}\left[\hat{P}_{XX}^{W}(f)\right]\right)^2 \tag{10.38}$$

Welch has shown that for the case when there is no overlap, $D = N$,

$$\mathrm{Var}[P_{XX}^{W}(f)] = \frac{\mathrm{Var}[P_{XX}^{(i)}(f)]}{K_1} \approx \frac{P_{XX}^2(f)}{K_1} \tag{10.39}$$

and for half-overlap, $D = N/2$,

$$\mathrm{Var}[\hat{P}_{XX}^{W}(f)] = \frac{9}{8K_2}P_{XX}^2(f) \tag{10.40}$$

10.4.4 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

$$\hat{P}_{XX}^{BT}(f) = \sum_{m=-(N-1)}^{N-1} w(m)\hat{r}_{xx}(m)e^{-j2\pi fm} \tag{10.41}$$

For a signal of N samples, the number of samples available for estimation of the autocorrelation value at the lag m, $\hat{r}_{xx}(m)$, decrease 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

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)] = \sum_{m=-(N-1)}^{N-1} \mathcal{E}[\hat{r}_{xx}(m)]w(m)e^{-j2\pi fm} \tag{10.42}$$

Now $\mathcal{E}[\hat{r}_{xx}(m)] = r_{xx}(m)w_B(m)$, where $w_B(m)$ is the Bartlett, or triangular, window. Equation (10.42) may be written as

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)] = \sum_{m=-(N-1)}^{N-1} r_{xx}(m)w_c(m)e^{-j2\pi fm} \tag{10.43}$$

where $w_c(m) = w_B(m)w(m)$. The right-hand side of Equation (10.43) can be written in terms of the Fourier transform of the autocorrelation and the window functions as

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)] = \int_{-1/2}^{1/2} P_{XX}(\nu)W_c(f-\nu)d\nu \tag{10.44}$$

where $W_c(f)$ is the Fourier transform of $w_c(m)$. The variance of the Blackman–Tukey estimate is given by

$$\text{Var}[\hat{P}_{XX}^{BT}(f)] \approx \frac{U}{N}P_{XX}^2(f) \tag{10.45}$$

where U is the energy of the window $w_c(m)$.

10.4.5 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 generalisation 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

$$x_i(m) = x(m+iD), \quad m = 0, 1, \ldots, 2N-1 \tag{10.46}$$
$$i = 0, 1, \ldots, K-1$$

where D is the overlap. For each segment of length $2N$, the correlation function in the range of $0 \geq m \geq N$ is given by

$$\hat{r}_{xx}(m) = \frac{1}{N}\sum_{k=0}^{N-1} x_i(k)x_i(k+m), \quad m = 0, 1, \ldots, N-1 \tag{10.47}$$

In Equation (10.47), the estimate of each correlation value is obtained as the averaged sum of N products.

10.5 Model-Based Power Spectrum Estimation

In non-parametric 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 generalised discrete linear time-invariant model is given by

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + \sum_{k=0}^{Q} b_k e(m-k) \tag{10.48}$$

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 (10.48) is known as an auto-regressive-moving-average (ARMA) model. The system function $H(z)$ of the discrete linear time-invariant model of Equation (10.48) is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum\limits_{k=0}^{Q} b_k z^{-k}}{1 - \sum\limits_{k=1}^{P} a_k z^{-k}} \tag{10.49}$$

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:

$$P_{XX}(f) = P_{EE}(f)\,|H(f)|^2 \tag{10.50}$$

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 (10.50) becomes

$$P_{XX}(f) = |H(f)|^2 \tag{10.51}$$

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 auto regressive (all-pole), a moving-average (all-zero) or an ARMA (pole–zero) model.

10.5.1 Maximum–Entropy Spectral Estimation

The power spectrum of a stationary signal is defined as the Fourier transform of the autocorrelation sequence:

$$P_{XX}(f) = \sum_{n=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \tag{10.52}$$

Equation (10.52) 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 non-parametric 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 a function of its bandwidth and spread in frequency domain and may be defined as

$$H\left[P_{XX}(f)\right] = \int_{-1/2}^{1/2} \ln P_{XX}(f)df \tag{10.53}$$

To obtain the maximum-entropy correlation estimate, we differentiate Equation (10.53) with respect to the unknown values of the correlation coefficients, and set the derivative to zero:

$$\frac{\partial H[P_{XX}(f)]}{\partial r_{xx}(m)} = \int_{-1/2}^{1/2} \frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)} df = 0 \quad \text{for } |m| > P \tag{10.54}$$

Now, from Equation (10.18), the derivative of the power spectrum with respect to the autocorrelation values is given by

$$\frac{\partial P_{XX}(f)}{\partial r_{xx}(m)} = e^{-j2\pi fm} \tag{10.55}$$

From Equation (10.55), for the derivative of the logarithm of the power spectrum, we have

$$\frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)} = P_{XX}^{-1}(f)e^{-j2\pi fm} \tag{10.56}$$

Substitution of Equation (10.56) in Equation (10.54) gives

$$\int_{-1/2}^{1/2} P_{XX}^{-1}(f)e^{-j2\pi fm}df = 0 \quad \text{for } |m| > P \tag{10.57}$$

Assuming that $P_{XX}^{-1}(f)$ is integrable, it may be associated with an autocorrelation sequence $c(m)$ as

$$P_{XX}^{-1}(f) = \sum_{m=-\infty}^{\infty} c(m)e^{-j2\pi fm} \tag{10.58}$$

where

$$c(m) = \int_{-1/2}^{1/2} P_{XX}^{-1}(f)e^{j2\pi fm}df \tag{10.59}$$

From Equations (10.57) and (10.59), we have $c(m) = 0$ for $|m| > P$. Hence, from Equation (10.58), the inverse of the maximum-entropy power spectrum may be obtained from the Fourier transform of a finite-length autocorrelation sequence as

$$P_{XX}^{-1}(f) = \sum_{m=-P}^{P} c(m)e^{-j2\pi fm} \tag{10.60}$$

and the maximum-entropy power spectrum is given by

$$\hat{P}_{XX}^{ME}(f)=\frac{1}{\sum\limits_{m=-P}^{P}c(m)e^{-j2\pi fm}} \tag{10.61}$$

Since the denominator polynomial in Equation (10.61) 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

$$\sum_{m=-P}^{P}c(m)z^{-m}=\frac{1}{\sigma^2}A(z)A(z^{-1}) \tag{10.62}$$

where $1/\sigma^2$ is a gain term, and $A(z)$ is a polynomial of order P defined as

$$A(z)=1+a_1z^{-1}+\cdots+a_pz^{-P} \tag{10.63}$$

From Equations (10.61) and (10.62), the maximum-entropy power spectrum may be expressed as

$$\hat{P}_{XX}^{ME}(f)=\frac{\sigma^2}{A(z)A(z^{-1})} \tag{10.64}$$

Equation (10.64) shows that the maximum-entropy power spectrum estimate is the power spectrum of an autoregressive (AR) model. Equation (10.64) 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 (10.64), as discussed in the next section.

10.5.2 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

$$x(m)=\sum_{k=1}^{P}a_kx(m-k)+e(m) \tag{10.65}$$

where $e(m)$ is a random signal of variance σ_e^2. The power spectrum of an autoregressive process is given by

$$P_{XX}^{AR}(f)=\frac{\sigma_e^2}{\left|1-\sum\limits_{k=1}^{P}a_ke^{-j2\pi fk}\right|^2} \tag{10.66}$$

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 (10.65) by $x(m-\mathrm{j})$ and taking the expectation:

$$\mathcal{E}[x(m)x(m-j)]=\sum_{k=1}^{P}a_k\mathcal{E}[x(m-k)x(m-j)]+\mathcal{E}[e(m)x(m-j)] \tag{10.67}$$

Now for the optimal model coefficients the random input $e(m)$ is orthogonal to the past samples, and Equation (10.67) becomes

$$r_{xx}(j)=\sum_{k=1}^{P}a_k\,r_{xx}(j-k),\quad j=1,2,\ldots \tag{10.68}$$

Given $P+1$ correlation values, Equation (10.68) can be solved to obtain the AR coefficients a_k. Equation (10.68) can also be used to extrapolate the correlation sequence. The methods of solving the AR model coefficients are discussed in Chapter 8.

10.5.3 Moving-Average Power Spectrum Estimation

A moving-average model is also known as an all-zero or a finite impulse response (FIR) filter. A signal $x(m)$, modelled as a moving-average process, is described as

$$x(m) = \sum_{k=0}^{Q} b_k \, e(m-k) \tag{10.69}$$

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

$$\begin{aligned} r_{xe}(m) &= \mathcal{E}[x(j)e(j-m)] \\ &= \mathcal{E}\left[\sum_{k=0}^{Q} b_k \, e(j-k)e(j-m)\right] = \sigma_e^2 b_m \end{aligned} \tag{10.70}$$

and the autocorrelation function of a moving average process is

$$r_{xx}(m) = \begin{cases} \sigma_e^2 \sum\limits_{k=0}^{Q-|m|} b_k b_{k+m}, & |m| \le Q \\ 0, & |m| > Q \end{cases} \tag{10.71}$$

From Equation (10.71), 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

$$P_{XX}^{MA} = \sum_{m=-Q}^{Q} r_{xx}(m) e^{-j2\pi fm} \tag{10.72}$$

Note that the moving-average spectral estimation is identical to the Blackman–Tukey method of estimating periodograms from the autocorrelation sequence.

10.5.4 Autoregressive Moving-Average Power Spectrum Estimation

The ARMA, or pole – zero, model is described by Equation (10.48). The relationship between the ARMA parameters and the autocorrelation sequence can be obtained by multiplying both sides of Equation (10.48) by $x(m-j)$ and taking the expectation:

$$r_{xx}(j) = \sum_{k=1}^{P} a_k r_{xx}(j-k) + \sum_{k=0}^{Q} b_k r_{xe}(j-k) \tag{10.73}$$

The moving-average part of Equation (10.73) influences the autocorrelation values only up to the lag of Q. Hence, for the autoregressive part of Equation (10.73), we have

$$r_{xx}(m) = \sum_{k=1}^{P} a_k r_{xx}(m-k) \quad \text{for} \quad m > Q \tag{10.74}$$

Hence Equation (10.74) can be used to obtain the coefficients a_k, which may then be substituted in Equation (10.73) for solving the coefficients b_k. Once the coefficients of an ARMA model are identified, the spectral estimate is given by

$$P_{XX}^{ARMA}(f) = \sigma_e^2 \frac{\left| \sum_{k=0}^{Q} b_k e^{-j2\pi fk} \right|^2}{\left| 1 - \sum_{k=1}^{P} a_k e^{-j2\pi fk} \right|^2} \tag{10.75}$$

where σ_e^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 anti-resonances of the spectrum.

10.6 High-Resolution Spectral Estimation Based on Subspace Eigen-Analysis

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. Eigen-analysis 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 eigen-analysis methods considered in this section.

10.6.1 Pisarenko Harmonic Decomposition

A real-valued sine wave can be modelled by a second-order autoregressive (AR) model, with its poles on the unit circle at the angular frequency of the sinusoid as shown in Figure 10.5. The AR model for a sinusoid of frequency F_i at a sampling rate of F_s is given by

$$x(m) = 2\cos(2\pi F_i/F_s)\, x(m-1) - x(m-2) + A\delta(m-t_0) \tag{10.76}$$

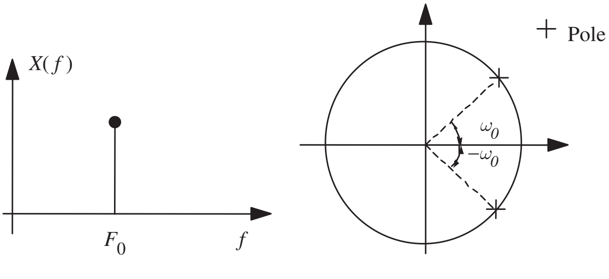


Figure 10.5 A second order all pole model of a sinusoidal signal.

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

$$x(m) = \sum_{k=1}^{2P} a_k x(m-k) + A\delta(m-t_0) \tag{10.77}$$

The transfer function of the AR model is given by

$$H(z) = \frac{A}{1 - \sum_{k=1}^{2P} a_k z^{-k}} = \frac{A}{\prod_{k=1}^{P} (1 - e^{-j2\pi F_k} z^{-1})(1 - e^{+j2\pi F_k} z^{-1})} \tag{10.78}$$

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

$$\begin{aligned} y(m) &= x(m) + n(m) \\ &= \sum_{k=1}^{2P} a_k x(m-k) + n(m) \end{aligned} \tag{10.79}$$

Substituting $[y(m-k) - n(m-k)]$ for $x(m-k)$ in Equation (10.79) yields

$$y(m) - \sum_{k=1}^{2P} a_k y(m-k) = n(m) - \sum_{k=1}^{2P} a_k n(m-k) \tag{10.80}$$

From Equation (10.80), 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 (10.80) can also be expressed in a vector notation as

$$\boldsymbol{y}^{\mathrm{T}} \boldsymbol{a} = \boldsymbol{n}^{\mathrm{T}} \boldsymbol{a} \tag{10.81}$$

where $\boldsymbol{y}^{\mathrm{T}} = [y(m), \ldots, y(m-2P)]$, $\boldsymbol{a}^{\mathrm{T}} = [1, a_1, \ldots, a_{2P}]$ and $\boldsymbol{n}^{\mathrm{T}} = [n(m), \ldots, n(m-2P)]$. To obtain the parameter vector $\boldsymbol{a}$, we multiply both sides of Equation (10.81) by the vector $\boldsymbol{y}$ and take the expectation:

$$\mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}]\boldsymbol{a} = \mathcal{E}[\boldsymbol{y}\boldsymbol{n}^{\mathrm{T}}]\boldsymbol{a} \tag{10.82}$$

or

$$\boldsymbol{R}_{yy}\boldsymbol{a} = \boldsymbol{R}_{yn}\boldsymbol{a} \tag{10.83}$$

where $\mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}] = \boldsymbol{R}_{yy}$, and $\mathcal{E}[\boldsymbol{y}\boldsymbol{n}^{\mathrm{T}}] = \boldsymbol{R}_{yn}$ can be written as

$$\begin{aligned} \boldsymbol{R}_{yn} &= \mathcal{E}[(\boldsymbol{x}+\boldsymbol{n})\boldsymbol{n}^{\mathrm{T}}] \\ &= \mathcal{E}[\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}] = \boldsymbol{R}_{nn} = \sigma_n^2 \mathbf{I} \end{aligned} \tag{10.84}$$

where σ_n^2 is the noise variance. Using Equation (10.84), Equation (10.83) becomes

$$\boldsymbol{R}_{yy}\boldsymbol{a} = \sigma_n^2 \boldsymbol{a} \tag{10.85}$$

Equation (10.85) is in the form of an eigen equation. If the dimension of the matrix $\boldsymbol{R}_{yy}$ 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 $\boldsymbol{a}$ is obtained as the eigenvector of $\boldsymbol{R}_{yy}$, with its first element unity and associated with the minimum eigenvalue. From the AR parameter vector $\boldsymbol{a}$, we can obtain the frequencies of the sinusoids by first calculating the roots of the polynomial

$$1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_2 z^{-2P+2} + a_1 z^{-2P+1} + z^{-2P} = 0 \tag{10.86}$$

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 (10.86) using the relation

$$z_k = e^{j2\pi F_k} \tag{10.87}$$

The powers of the sinusoids are calculated as follows. For P sinusoids observed in additive white noise, the autocorrelation function is given by

$$r_{yy}(k) = \sum_{i=1}^{P} P_i \cos 2k\pi F_i + \sigma_n^2 \delta(k) \tag{10.88}$$

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 (10.88) for the correlation lags $k = 1, \ldots, P$ can be written as

$$\begin{pmatrix} \cos 2\pi F_1 & \cos 2\pi F_2 & \cdots & \cos 2\pi F_P \\ \cos 4\pi F_1 & \cos 4\pi F_2 & \cdots & \cos 4\pi F_P \\ \vdots & \vdots & \ddots & \vdots \\ \cos 2P\pi F_1 & \cos 2P\pi F_2 & \cdots & \cos 2P\pi F_P \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_P \end{pmatrix} = \begin{pmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \vdots \\ r_{yy}(P) \end{pmatrix} \tag{10.89}$$

Given an estimate of the frequencies F_i from Equations (10.86) and (10.87), and an estimate of the autocorrelation function $\hat{r}_{yy}(k)$, Equation (10.89) can be solved to obtain the powers of the sinusoids P_i. The noise variance can then be obtained from Equation (10.88) as

$$\sigma_n^2 = r_{yy}(0) - \sum_{i=1}^{P} P_i \tag{10.90}$$

10.6.2 Multiple Signal Classification (MUSIC) Spectral Estimation

The 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

$$y(m) = \sum_{k=1}^{P} A_k e^{-\mathrm{j}(2\pi F_k m + \phi_k)} + n(m) \tag{10.91}$$

An N-sample vector $\boldsymbol{y} = [y(m), \ldots, y(m+N-1)]$ of the noisy signal can be written as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} + \boldsymbol{n} \\ &= \boldsymbol{Sa} + \boldsymbol{n} \end{aligned} \tag{10.92}$$

where the signal vector $\boldsymbol{x} = \boldsymbol{Sa}$ is defined as

$$\begin{pmatrix} x(m) \\ x(m+1) \\ \vdots \\ x(m+N-1) \end{pmatrix} = \begin{pmatrix} e^{\mathrm{j}2\pi F_1 m} & e^{\mathrm{j}2\pi F_2 m} & \cdots & e^{\mathrm{j}2\pi F_P m} \\ e^{\mathrm{j}2\pi F_1 (m+1)} & e^{\mathrm{j}2\pi F_2 (m+1)} & \cdots & e^{\mathrm{j}2\pi F_P (m+1)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\mathrm{j}2\pi F_1 (m+N-1)} & e^{\mathrm{j}2\pi F_2 (m+N-1)} & \cdots & e^{\mathrm{j}2\pi F_P (m+N-1)} \end{pmatrix} \begin{pmatrix} A_1 e^{\mathrm{j}2\pi\phi_1} \\ A_2 e^{\mathrm{j}2\pi\phi_2} \\ \vdots \\ A_P e^{\mathrm{j}2\pi\phi_P} \end{pmatrix} \tag{10.93}$$

The matrix $\boldsymbol{S}$ and the vector $\boldsymbol{a}$ are defined on the right-hand side of Equation (10.93). The autocorrelation matrix of the noisy signal $\boldsymbol{y}$ can be written as the sum of the autocorrelation matrices of the signal $\boldsymbol{x}$ and the noise as

$$\begin{aligned} \boldsymbol{R}_{yy} &= \boldsymbol{R}_{xx} + \boldsymbol{R}_{nn} \\ &= \boldsymbol{SPS}^{\mathrm{H}} + \sigma_n^2 \mathbf{I} \end{aligned} \tag{10.94}$$

where $\boldsymbol{R}_{xx} = \boldsymbol{SPS}^{\mathrm{H}}$ and $\boldsymbol{R}_{nn} = \sigma_n{}^2\mathbf{I}$ are the autocorrelation matrices of the signal and noise processes, the exponent H denotes the Hermitian transpose, and the diagonal matrix $\boldsymbol{P}$ defines the power of the sinusoids as

$$\boldsymbol{P} = \boldsymbol{aa}^{\mathrm{H}} = \mathrm{diag}[P_1, P_2, \ldots, P_P] \tag{10.95}$$

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

$$\boldsymbol{R}_{xx} = \sum_{k=1}^{P} P_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{H}} \tag{10.96}$$

where $\boldsymbol{s}_k^{\mathrm{H}} = [1, e^{j2\pi F_k}, \cdots, e^{j2\pi(N-1)F_k}]$. Now consider an eigen-decomposition of the $N \times N$ correlation matrix $\boldsymbol{R}_{xx}$

$$\begin{aligned}\boldsymbol{R}_{xx} &= \sum_{k=1}^{N} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \\ &= \sum_{k=1}^{P} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}}\end{aligned} \tag{10.97}$$

where λ_k and $\boldsymbol{v}_k$ are the eigenvalues and eigenvectors of the matrix $\boldsymbol{R}_{xx}$ respectively. We have also used the fact that the autocorrelation matrix $\boldsymbol{R}_{xx}$ of P complex sinusoids has only P non-zero eigenvalues, $\lambda_{P+1} = \lambda_{P+2}, \ldots, \lambda_N = 0$. 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 $\boldsymbol{R}_{xx}$ as

$$\boldsymbol{R}_{nn} = \sigma_n^2 \mathbf{I} = \sigma_n^2 \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \tag{10.98}$$

The correlation matrix of the noisy signal may be expressed in terms of its eigenvectors and the associated eigenvalues of the noisy signal as

$$\begin{aligned}\boldsymbol{R}_{yy} &= \sum_{k=1}^{P} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} + \sigma_n^2 \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \\ &= \sum_{k=1}^{P} \left(\lambda_k + \sigma_n^2\right) \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} + \sigma_n^2 \sum_{k=P+1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}}\end{aligned} \tag{10.99}$$

From Equation (10.99), the eigenvectors and the eigenvalues of the correlation matrix of the noisy signal can be partitioned into two disjoint subsets (see Figure 10.6). The set of eigenvectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_P\}$, associated with the P largest eigenvalues span the *signal subspace* and are called the *principal eigenvectors*.

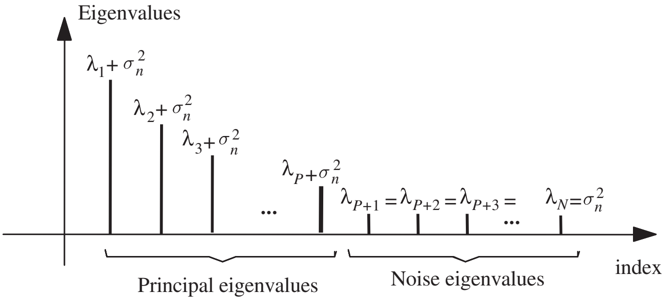


Figure 10.6 Decomposition of the eigenvalues of a noisy signal into the principal eigenvalues and the noise eigenvalues.

The signal vectors $\boldsymbol{s}_i$ can be expressed as linear combinations of the principal eigenvectors. The second subset of eigenvectors $\{\boldsymbol{v}_{P+1}, \ldots, \boldsymbol{v}_N\}$ span the *noise subspace* and 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 $\boldsymbol{s}_i$ which are in the signal subspace, are orthogonal to the noise subspace, and we have

$$\boldsymbol{s}_i^{\mathrm{H}}(f)\boldsymbol{v}_k = \sum_{m=0}^{N-1} v_k(m)e^{-j2\pi F_i m} = 0 \quad i = 1, \ldots, P \quad k = P+1, \ldots, N \tag{10.100}$$

Equation (10.100) 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:

$$\sum_{k=P+1}^{N} \boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k \tag{10.101}$$

In the MUSIC algorithm, the power spectrum estimate is defined as

$$P_{XX}(f) = \sum_{k=P+1}^{N} \left|\boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k\right|^2 \tag{10.102}$$

where $\boldsymbol{s}(f) = [1, \; e^{j2\pi f}, \; \ldots, \; e^{j2\pi(N-1)f}]$ is the complex sinusoidal vector, and $\{\boldsymbol{v}_{P+1}, \; \ldots, \; \boldsymbol{v}_N\}$ are the eigenvectors in the noise subspace. From Equations (10.10) and (10.10) we have that

$$P_{XX}(f_i) = 0, \quad i = 1, \ldots, P \tag{10.103}$$

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

$$P_{XX}^{MUSIC}(f) = \frac{1}{\sum\limits_{k=P+1}^{N} |\boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k|^2} = \frac{1}{\boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{V}(f)\boldsymbol{V}^{\mathrm{H}}(f)\boldsymbol{s}(f)} \tag{10.104}$$

where $\boldsymbol{V} = [\boldsymbol{v}_{P+1}, \ldots, \boldsymbol{v}_N]$ is the matrix of eigenvectors of the noise subspace. $P_{MUSIC}(f)$ 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.

10.6.3 Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT)

The ESPIRIT algorithm is an eigen-decomposition 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:

$$y(m) = \sum_{k=1}^{P} A_k e^{-j(2\pi F_k m + \phi_k)} + n(m) \tag{10.105}$$

The ESPIRIT algorithm exploits the deterministic relation between sinusoidal component of the signal vector $\boldsymbol{y}(m) = [y(m), \; \ldots, \; y(m+N-1)]^{\mathrm{T}}$ and that of the time-shifted vector $\boldsymbol{y}(m+1) = [y(m+1), \; \ldots, \; y(m+N)]^{\mathrm{T}}$. The signal component of the noisy vector $\boldsymbol{y}(m)$ may be expressed as

$$\boldsymbol{x}(m) = \boldsymbol{S}\,\boldsymbol{a} \tag{10.106}$$

where $\boldsymbol{S}$ is the complex sinusoidal matrix and $\boldsymbol{a}$ is the vector containing the amplitude and phase of the sinusoids as in Equations (10.92) and (10.93). 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 $\boldsymbol{x}(m+1)$ may be obtained from $\boldsymbol{x}(m)$ by phase-shifting each complex sinusoidal component of $\boldsymbol{x}(m)$ as

$$\boldsymbol{x}(m+1) = \boldsymbol{S\Phi a} \tag{10.107}$$

where $\boldsymbol{\Phi}$ is a $P \times P$ phase matrix defined as

$$\boldsymbol{\Phi} = \mathrm{diag}[e^{j2\pi F_1}, e^{j2\pi F_2}, \cdots, e^{j2\pi F_P}] \tag{10.108}$$

The diagonal elements of $\boldsymbol{\Phi}$ are the relative phases between the adjacent samples of the sinusoids. The matrix $\boldsymbol{\Phi}$ is a unitary matrix and is known as a *rotation matrix* since it relates the time-shifted vectors $\boldsymbol{x}(m)$ and $\boldsymbol{x}(m+1)$. The autocorrelation matrix of the noisy signal vector $\boldsymbol{y}(m)$ can be written as

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m)} = \boldsymbol{SPS}^{\mathrm{H}} + \sigma_n^2\mathbf{I} \tag{10.109}$$

where the matrix $\boldsymbol{P}$ is diagonal, and its diagonal elements are the powers of the complex sinusoids $\boldsymbol{P} = \mathrm{diag}[A_1^2, \ldots, A_P^2] = \boldsymbol{aa}^{\mathrm{H}}$. The cross-covariance matrix of the vectors $\boldsymbol{y}(m)$ and $\boldsymbol{y}(m+1)$ is

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} = \boldsymbol{SP\Phi}^{\mathrm{H}}\boldsymbol{S}^{\mathrm{H}} + \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} \tag{10.110}$$

where the autocovariance matrices $\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)}$ and $\boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)}$ are defined as

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} = \begin{pmatrix} \boldsymbol{r}_{yy}(1) & \boldsymbol{r}_{yy}(2) & \boldsymbol{r}_{yy}(3) & \cdots & \mathbf{r}_{\mathrm{yy}}(N) \\ \boldsymbol{r}_{yy}(0) & \boldsymbol{r}_{yy}(1) & \boldsymbol{r}_{yy}(2) & \cdots & \boldsymbol{r}_{yy}(N-1) \\ \boldsymbol{r}_{yy}(1) & \boldsymbol{r}_{yy}(0) & \boldsymbol{r}_{yy}(1) & \cdots & \boldsymbol{r}_{yy}(N-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{r}_{yy}(N-2) & \boldsymbol{r}_{yy}(N-3) & \boldsymbol{r}_{yy}(N-4) & \cdots & \boldsymbol{r}_{yy}(1) \end{pmatrix} \tag{10.111}$$

and

$$\boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \sigma_n^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_n^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 & 0 \end{pmatrix} \tag{10.112}$$

The correlation matrix of the signal vector $\boldsymbol{x}(m)$ can be estimated as

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m)} = \boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m)} - \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m)} = \boldsymbol{SPS}^{\mathrm{H}} \tag{10.113}$$

and the cross-correlation matrix of the signal vector $\boldsymbol{x}(m)$ with its time-shifted version $\boldsymbol{x}(m+1)$ is obtained as

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m+1)} = \boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} - \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} = \boldsymbol{SP\Phi}^{\mathrm{H}}\boldsymbol{S}^{\mathrm{H}} \tag{10.114}$$

Subtraction of a fraction $\lambda_i = e^{-j2\pi F_i}$ of Equation (10.114) from Equation (10.113) yields

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m)} - \lambda_i\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m+1)} = \boldsymbol{SP}(\mathbf{I} - \lambda_i\boldsymbol{\Phi}^H)\boldsymbol{S}^H \tag{10.115}$$

From Equations (10.108) and (10.115), the frequencies of the sinusoids can be estimated as the roots of Equation (10.115).

10.7 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 10.1 on basic definitions of the Fourier series/transform, energy spectrum and power spectrum. In Section 10.2, we considered non-parametric 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 10.3, 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 less side-lobes. In Section 10.4, we considered the eigen-based spectral estimation of noisy signals. These methods decompose the eigen variables 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. *Biometrica*. **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 Publications, 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. PhD Thesis, Department of Geophysics, Stanford University, California.

Cadzow J.A. (1979) ARMA Spectral Estimation: An Efficient Closed-form Procedure. Proc. RADC Spectrum estimation Workshop: 81–97.

Capon J. (1969) High Resolution Frequency-Wavenumber Spectrum Analysis. *Proc. IEEE*. **57**: 1408–1419.

Childers D.G., Editor (1978) *Modern Spectrum Analysis*. IEEE Press.

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. Mathematics of Computation, **19, 90**: 297–301.

Fourier, J.B.J. (1878) *Théorie Analytique de la Chaleur*, Trans. Alexander Freeman; Repr. Dover Publications, 1955.

Grattam-Guiness I. (1972) *Joseph Fourier (1768–1830): A Survey of His Life and Work*. MIT Press.

Haykin S. (1985) *Array Signal Processing*. Prentice-Hall, NJ.

Jenkins G.M. and Watts D.G. (1968) *Spectral Analysis and Its Applications*. Holden-Day, San Francisco, California.

Kay S.M. and Marple S.L. (1981) Spectrum Analysis: A Modern Perspective. *Proc. IEEE*, **69**: 1380–1419.

Kay S.M. (1988) *Modern Spectral Estimation: Theory and Application*. Prentice Hall-Englewood Cliffs, NJ.

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. Stat.*, **28**: 329–349.

Pisarenko V.F. (1973) The Retrieval of Harmonics from a Covariance Function. *Geophy. J.R. Astron. Soc.*, **33**: 347–366.

Roy R.H. (1987) ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. PhD Thesis, Stanford University, California.

Schmidt R.O. (1981) A signal Subspace Approach to Multiple Emitter Location and Spectral Estimation. PhD Thesis, Stanford University, California.

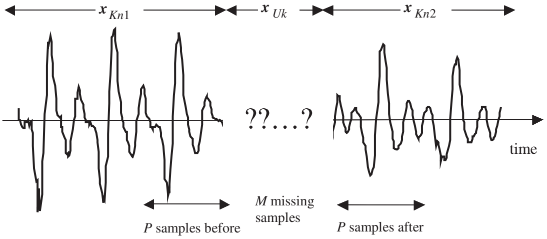
Stanislav B.K., Ed (1986) *Modern Spectrum Analysis*. IEEE Press.

Strand O.N. (1977) Multichannel Complex Maximum Entropy (AutoRegressive) Spectral Analysis. *IEEE Trans. Automatic Control*, **22**(4): 634–640.

Van Den Bos A. (1971) Alternative Interpretation of Maximum Entropy Spectral Analysis. *IEEE Trans. Infor. Tech.*, **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 and Electroacoustics*, **AU-15**: 70–79.

Wilkinson J.H. (1965) *The Algebraic Eigenvalue Problem*. Oxford, Oxford University Press.



11

Interpolation – Replacement of Lost Samples

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 neighborhood points. Interpolators are used in various forms in most communication 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 communication 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 11.2, 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 11.3, 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.

11.1 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 on the effects of a sequence of lost samples and on the methods of recovery of the lost samples. The factors that affect interpolation error are also considered in this section.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

11.1.1 Ideal 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:

$$x(m) = x(t)p(t) = \sum_{m=-\infty}^{\infty} x(t)\delta(t - mT_s) \tag{11.1}$$

where $p(t) = \Sigma\delta(t - mT_s)$ is the sampling function and $T_s = 1/F_s$ is the sampling interval. Taking the Fourier transform of Equation (11.1), it can be shown that the spectrum of the sampled signal is given by

$$X_s(f) = X(f) * P(f) = \sum_{k=-\infty}^{\infty} X(f + kF_s) \tag{11.2}$$

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 (11.2), illustrated in Figure 11.1, 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 11.1. Hence the ideal interpolator of a band-limited discrete-time signal is an ideal low-pass filter with a sinc impulse response. The recovery of a continuous-time signal through sinc interpolation can be expressed as

$$x(t) = \sum_{m=-\infty}^{\infty} x(m)T_sF_c sinc[\pi f_c(t - mT_s)] \tag{11.3}$$

In practice, the sampling rate F_s should be sufficiently greater than $2F_c$, say $2.2F_c$, in order to accommodate the transition bandwidth of the interpolating low-pass filter.

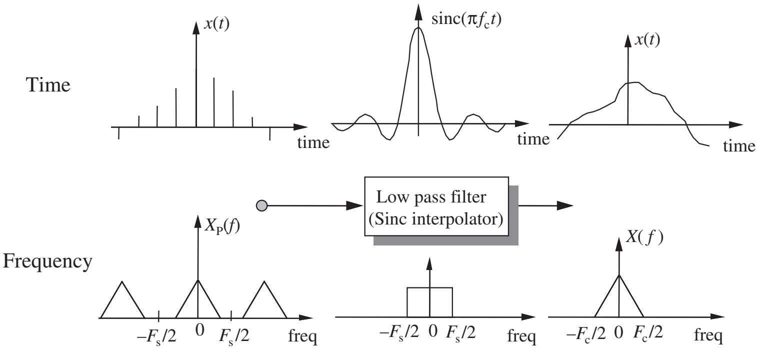


Figure 11.1 Reconstruction of a continuous-time signal from its discrete-time samples. In frequency domain interpolation is equivalent to low-pass filtering.

11.1.2 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.

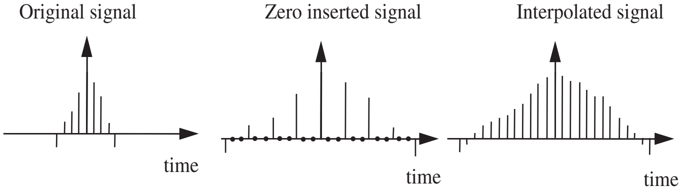


Figure 11.2 Illustration of up-sampling by a factor of 3, using a two-stage process of zero-insertion and digital low-pass filtering.

Consider a band-limited discrete-time signal $x(m)$ with a base-band spectrum $X(f)$ as shown in Figure 11.2. 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 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

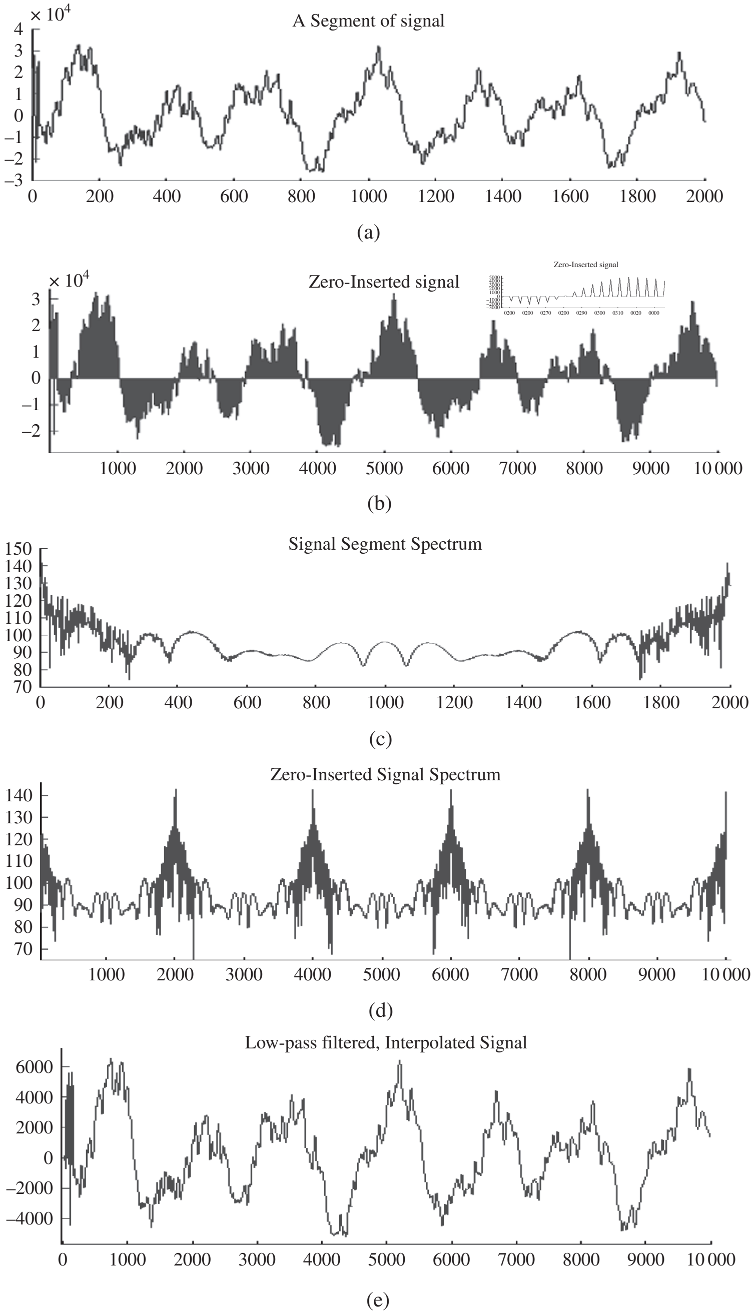
$$x_z(m) = \begin{cases} x\left(\frac{m}{I}\right), & m = 0, \pm I, \pm 2I, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{11.4}$$

The spectrum of the zero-inserted signal is related to the spectrum of the original discrete-time signal by

$$\begin{aligned} X_z(f) &= \sum_{m=-\infty}^{\infty} x_z(m) e^{-j2\pi fm} \\ &= \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi fmI} \\ &= X(I.f) \end{aligned} \tag{11.5}$$

Equation (11.5) 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 11.3. A segment of speech and its zero-inserted version are shown in Figures 11.3(a) and 11.3(b) respectively. Figures 11.3(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 11.3(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.



(a)

(b)

(c)

(d)

(e)

Figure 11.3 (a) Original signal, (b) zero-inserted signal, (c) spectrum of original signal, (d) spectrum of zero-inserted signal, (e) interpolated signal.

11.1.3 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 side of the gap, as illustrated in Figure 11.4. 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, 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.

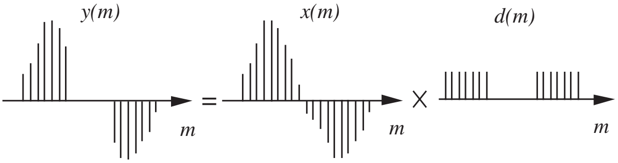


Figure 11.4 Illustration of a distortion model for a signal with a sequence of missing samples.

A simple distortion model for a signal $y(m)$ with M missing samples, illustrated in Figure 11.4, is given by

$$\begin{aligned} y(m) &= x(m)d(m) \\ &= x(m)[1 - r(m)] \end{aligned} \tag{11.6}$$

where the distortion operator $d(m)$ is defined as

$$d(m) = 1 - r(m) \tag{11.7}$$

and $r(m)$ is a rectangular pulse of duration M samples starting at the sampling time k:

$$r(m) = \begin{cases} 1, & k \le m \le k + M - 1 \\ 0, & \text{otherwise} \end{cases} \tag{11.8}$$

In the frequency domain, Equation (11.6) becomes

$$\begin{aligned} Y(f) &= X(f) * D(f) \\ &= X(f) * [\delta(f) - R(f)] \\ &= X(f) - X(f) * R(f) \end{aligned} \tag{11.9}$$

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

$$R(f) = e^{-j2\pi f(k+(M-1)/2)} \frac{\sin(\pi f M)}{\sin(\pi f)} \tag{11.10}$$

In general, the distortion $d(m)$ is a non-invertible, many-to-one transformation, and perfect interpolation with zero error is not possible. However, as discussed in Section 11.3, the interpolation error can be minimised through optimal utilisation of the signal models and the information contained in the neighbouring samples.

Example 11.1 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

$$\begin{aligned} y(m) &= x(m)\ d(m) \\ &= A(\cos 2\pi F_0 m)[1 - r(m)] \end{aligned} \tag{11.11}$$

where $r(m)$ is the rectangular pulse defined in Equation (11.8). In the frequency domain, the distorted signal can be expressed as

$$\begin{aligned} Y(f) &= \frac{A}{2}[\delta(f - F_o) + \delta(f + F_o)] * [\delta(f) - R(f)] \\ &= \frac{A}{2}\,[\delta\,(f - F_o) + \delta(f + F_o) - R(f - F_o) - R(f + F_o)] \end{aligned} \tag{11.12}$$

where $R(f)$ is the spectrum of the pulse $r(m)$ as in Equation (11.9).

From Equation (11.12), 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.

11.1.4 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 the increasing correlation structure, or equivalently the 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) Non-stationary 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 make efficient use of a relatively large number of known samples, or to interpolate a relatively large segment of missing samples.

In Section 11.3, 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.

11.2 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 $\{x(t_0), x(t_1), \ldots, x(t_\mathrm{N})\}$, 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 11.5) and may be written in power series form as

$$\hat{x}(t) = p_N(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_N t^N \tag{11.13}$$

where $P_N(t)$ is a polynomial of order N, and the a_k are the polynomial coefficients. From Equation (11.13), and a set of $N+1$ known samples, a system of $N+1$ linear equations with $N+1$ unknown coefficients can be formulated as

$$\begin{aligned} x(t_0) &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + \cdots + a_N t_0^N \\ x(t_1) &= a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + \cdots + a_N t_1^N \\ \vdots \quad & \quad \vdots \qquad\qquad \ddots \\ x(t_N) &= a_0 + a_1 t_N + a_2 t_N^2 + a_3 t_N^3 + \cdots + a_N t_N^N \end{aligned} \tag{11.14}$$

From Equation (11.14) the polynomial coefficients are given by

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 & \cdots & t_0^N \\ 1 & t_1 & t_1^2 & t_1^3 & \cdots & t_1^N \\ 1 & t_2 & t_2^2 & t_2^3 & \cdots & t_2^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 & \cdots & t_N^N \end{pmatrix}^{-1} \begin{pmatrix} x(t_0) \\ x(t_1) \\ x(t_2) \\ \vdots \\ x(t_N) \end{pmatrix} \tag{11.15}$$

The matrix in Equation (11.15) 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.

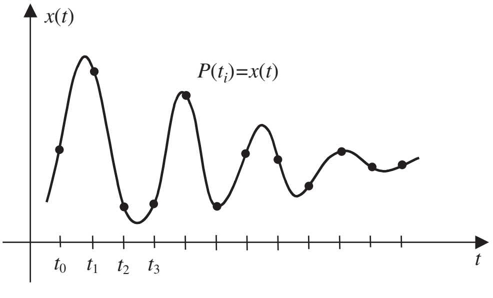


Figure 11.5 Illustration of an Interpolation curve through a number of samples.

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.

11.2.1 Lagrange Polynomial Interpolation

To introduce the Lagrange interpolation, consider a line interpolator passing through two points $x(t_0)$ and $x(t_1)$:

$$\hat{x}(t) = p_1(t) = x(t_0) + \underbrace{\frac{x(t_1) - x(t_0)}{t_1 - t_0}}_{\text{line slope}}(t - t_0) \tag{11.16}$$

The line Equation (11.16) may be rearranged and expressed as

$$p_1(t) = \frac{t - t_1}{t_0 - t_1} x(t_0) + \frac{t - t_0}{t_1 - t_0} x(t_1) \tag{11.17}$$

Equation (11.17) 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 11.6.

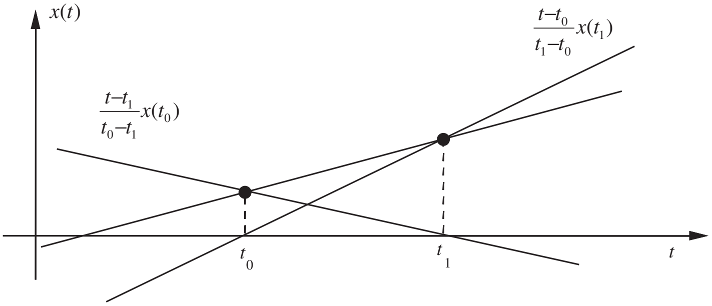


Figure 11.6 The Lagrange line interpolator passing through $x(t_0)$ and $x(t_1)$ described in terms of the combination of two lines: one passing through $(x(t_0),\ t_1)$ and the other through $(x(t_1),\ t_0)$.

In general, the Lagrange polynomial, of order N, passing through $N + 1$ samples $\{x(t_0), x(t_1), \ldots x(t_\mathrm{N})\}$ is given by the polynomial equation

$$P_N(t) = L_0(t)x(t_0) + L_1(t)x(t_1) + \cdots + L_N(t)x(t_N) \tag{11.18}$$

where each Lagrange coefficient $L_N(t)$ is itself a polynomial of degree N given by

$$L_i(t) = \frac{(t - t_0) \cdots (t - t_{i-1})(t - t_{i+1}) \cdots (t - t_N)}{(t_i - t_0) \cdots (t_i - t_{i-1})(t_i - t_{i+1}) \cdots (t_i - t_N)} = \prod_{\substack{n=0 \\ n \neq i}}^{N} \frac{t - t_n}{t_i - t_n} \tag{11.19}$$

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.

11.2.2 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:

$$
\begin{aligned}
p_0(t) &= a_0 && \text{(d.c. value)}\\
p_1(t) &= a_0 + a_1(t-t_0) && \text{(ramp)}\\
&= p_0(t) + a_1(t-t_0)\\
p_2(t) &= \underbrace{a_0 + a_1(t-t_0)}_{p_1(t)} + a_2(t-t_0)(t-t_1) && \text{(quadratic)}\\
&= p_1(t) + a_2(t-t_0)(t-t_1)\\
p_3(t) &= \underbrace{a_0 + a_1(t-t_0) + a_2(t-t_0)(t-t_1)}_{p_2(t)} + a_3(t-t_0)(t-t_1)(t-t_2) && \text{(cubic)}\\
&= p_2(t) + a_3(t-t_0)(t-t_1)(t-t_2)
\end{aligned}
\tag{11.20}
$$

and in general the recursive, *order update*, form of a Newton polynomial can be formulated as

$$p_N(t) = p_{N-1}(t) + a_N(t-t_0)(t-t_1)\cdots(t-t_{N-1}) \tag{11.21}$$

For a sequence of $N+1$ samples $\{x(t_0),\ x(t_1), \ldots x(t_N)\}$, 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 (11.21) at $t = t_0$ to $x(t_0)$:

$$p_N(t_0) = p_0(t_0) = x(t_0) = a_0 \tag{11.22}$$

To solve for the coefficient a_1, the first-order polynomial $p_1(t)$ is evaluated at $t = t_1$:

$$p_1(t_1) = x(t_1) = a_0 + a_1(t_1 - t_0) = x(t_0) + a_1(t_1 - t_0) \tag{11.23}$$

from which

$$a_1 = \frac{x(t_1) - x(t_0)}{t_1 - t_0} \tag{11.24}$$

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 $t = t_2$:

$$p_2(t_2) = x(t_2) = a_0 + a_1(t_2 - t_0) + a_2(t_2 - t_0)(t_2 - t_1) \tag{11.25}$$

Substituting a_0 and a_1 from Equations (11.22) and (11.24) in Equation (11.25) we obtain

$$a_2 = \left[\frac{x(t_2) - x(t_1)}{t_2 - t_1} - \frac{x(t_1) - x(t_0)}{t_1 - t_0}\right] \Bigg/ (t_2 - t_0) \tag{11.26}$$

Each term in the square brackets of Equation (11.26) 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 (11.26) is a so-called 'divided difference'. The divided difference between two points t_i and t_{i-1} is defined as

$$d_1(t_{i-1}, t_i) = \frac{x(t_i) - x(t_{i-1})}{t_i - t_{i-1}} \tag{11.27}$$

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

$$d_2(t_{i-2}, t_i) = \frac{d_1(t_{i-1}, t_i) - d_1(t_{i-2}, t_{i-1})}{t_i - t_{i-2}} \tag{11.28}$$

and the third-order divided difference is

$$d_3(t_{i-3}, t_i) = \frac{d_2(t_{i-2}, t_i) - d_2(t_{i-3}, t_{i-1})}{t_i - t_{i-3}} \tag{11.29}$$

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

$$d_j(t_{i-j}, t_i) = \frac{d_{j-1}(t_{i-j+1}, t_i) - d_{j-1}(t_{i-j}, t_{i-1})}{t_i - t_{i-j}} \tag{11.30}$$

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

$$a_i = d_i(t_0, t_i) \tag{11.31}$$

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.

11.2.3 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 $N+1$ sample 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 $[x(t_i), x'(t_i), x''(t_i), \ldots, x^{(M)}(t_i), i = 0, \ldots, N]$. There are altogether $K = (N+1)(M+1)$ data points and a polynomial of order $K-1$ can be fitted to the data as

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_{K-1} t^{K-1} \tag{11.32}$$

To obtain the polynomial coefficients, we substitute the given samples in the polynomial and its M derivatives as

$$\begin{aligned} p(t_i) &= x(t_i) \\ p'(t_i) &= x'(t_i) \\ p''(t_i) &= x''(t_i) \\ \vdots &= \vdots \\ p^{(M)}(t_i) &= x^{(M)}(t_i), \quad i = 0, 1, \ldots, N \end{aligned} \tag{11.33}$$

In all, there are $K = (M+1)(N+1)$ equations in (11.33), and these can be used to calculate the coefficients of the polynomial Equation (11.32). 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, the size of Equation (11.33) becomes too large for most practical purposes.

11.2.4 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

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{11.34}$$

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 $t_i \le t \le t_{i+1}$ of length $T_i = t_{i+1} - t_i$ as shown in Figure 11.7. Using a local coordinate $\tau = t - t_i$, the cubic polynomial becomes

$$p(\tau) = a_0 + a_1 \tau + a_2 \tau^2 + a_3 \tau^3 \tag{11.35}$$

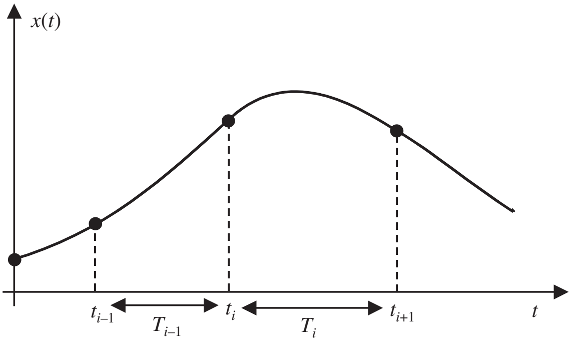


Figure 11.7 Illustration of cubic spline interpolation.

At $\tau = 0$, we obtain the first coefficient a_0 as

$$a_0 = p(\tau = 0) = x(t_i) \tag{11.36}$$

The second derivative of $p(\tau)$ is given by

$$p''(\tau) = 2a_2 + 6a_3\tau \tag{11.37}$$

Evaluation of the second derivative at $\tau = 0$ (i.e. $t = t_i$) gives the coefficient a_2

$$a_2 = \frac{p_i''(\tau = 0)}{2} = \frac{p_i''}{2} \tag{11.38}$$

Similarly, evaluating the second derivative at the point t_{i+1} (i.e. $\tau = T_i$) yields the fourth coefficient

$$a_3 = \frac{p_{i+1}'' - p_i''}{6T_i} \tag{11.39}$$

Now to obtain the coefficient a_1, we evaluate $p(\tau)$ at $\tau = T_i$:

$$p(\tau = T_i) = a_0 + a_1 T_i + a_2 T_i^2 + a_3 T_i^3 = x(t_{i+1}) \tag{11.40}$$

and substitute a_0, a_2 and a_3 from Equations (11.36), (11.38) and (11.39) in (11.40) to obtain

$$a_1 = \frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p''_{i+1} + 2p''_i}{6} T_i \tag{11.41}$$

The cubic polynomial can now be written as

$$p(\tau) = x(t_i) + \left[\frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p''_{i+1} + 2p''_i}{6} T_i\right]\tau + \frac{p''_i}{2}\tau^2 + \frac{p''_{i+1} - p''_i}{6T_i}\tau^3 \tag{11.42}$$

To determine the coefficients of the polynomial in Equation (11.42), 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 (11.42), the first derivatives of $p(\tau)$ evaluated at the endpoints t_i and t_{i+1} are

$$p'_i = p'(\tau = 0) = -\frac{T_i}{6}\left[p''_{i+1} + 2p''_i\right] + \frac{1}{T_i}\left[x(t_{i+1}) - x(t_i)\right] \tag{11.43}$$

$$p'_{i+1} = p'(\tau = T_i) = \frac{T_i}{6}\left[2p''_{i+1} + p''_i\right] + \frac{1}{T_i}\left[x(t_{i+1}) - x(t_i)\right] \tag{11.44}$$

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

$$p'_i = p'(\tau = t_i) = \frac{T_{i-1}}{6}\left[2p''_i + p''_{i-1}\right] + \frac{1}{T_{i-1}}\left[x(t_i) - x(t_{i-1})\right] \tag{11.45}$$

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 (11.43) and (11.45) and repeating this exercise yields

$$T_{i-1}p''_{i-1} + 2(T_{i-1} + T_i)p''_i + T_i p''_{i+1} = 6\left[\frac{1}{T_{i-1}}x(t_{i-1}) - \left(\frac{1}{T_{i-1}} + \frac{1}{T_i}\right)x(t_i) + \frac{1}{T_i}x(t_{i+1})\right]$$
$$i = 1, 2, \ldots, N-1 \tag{11.46}$$

In Equation (11.46), 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.

11.3 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 11.8 and 11.9 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, $\boldsymbol{x}_{\mathrm{Uk}} = \{x(k), x(k+1), \ldots, x(k+M-1)\}$ are missing. The objective is to make an optimal estimate of the missing segment $\boldsymbol{x}_{\mathrm{Uk}}$, using the remaining $N - k$ samples $\boldsymbol{x}_{\mathrm{Kn}}$ and a model of the signal

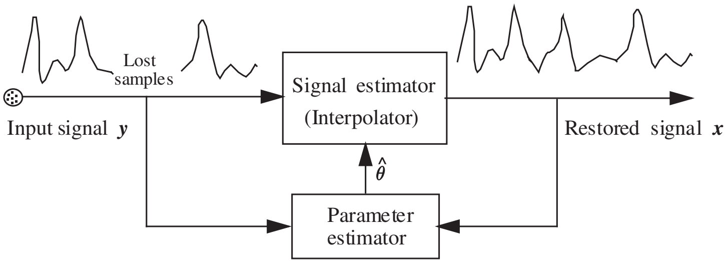


Figure 11.8 Illustration of a model-based iterative signal interpolation system.

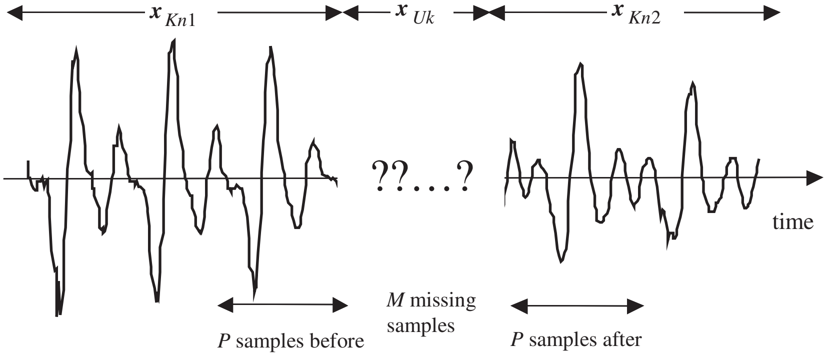


Figure 11.9 A signal with M missing samples and $N-M$ known samples. On each side of the missing segment, P samples are used to interpolate the segment.

process. An N-sample signal vector $\boldsymbol{x}$, composed of M unknown samples and $N-M$ known samples, can be written as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_{Kn_1} \\ \boldsymbol{x}_U \\ \boldsymbol{x}_{Kn_2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{Kn_1} \\ \boldsymbol{0} \\ \boldsymbol{x}_{Kn_2} \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{x}_{Uk} \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{K}\,\boldsymbol{x}_{Kn} + \boldsymbol{U}\,\boldsymbol{x}_{Uk} \tag{11.47}$$

where the vector $\boldsymbol{x}_{\mathrm{Kn}} = [\boldsymbol{x}_{\mathrm{Kn1}}\ \boldsymbol{x}_{\mathrm{Kn2}}]^{\mathrm{T}}$ is composed of the known samples, and the vector $\boldsymbol{x}_{\mathrm{Uk}}$ is composed of the unknown samples, as illustrated in Figure 11.8. The matrices $\boldsymbol{K}$ and $\boldsymbol{U}$ in Equation (11.47) are rearrangement matrices that assemble the vector $\boldsymbol{x}$ from $\boldsymbol{x}_{\mathrm{Kn}}$ and $\boldsymbol{x}_{\mathrm{Uk}}$.

11.3.1 Maximum A Posteriori Interpolation

The posterior pdf of an unknown signal segment $\boldsymbol{x}_{\mathrm{Uk}}$ given a number of neighbouring samples $\boldsymbol{x}_{\mathrm{Kn}}$ can be expressed using Bayes' rule as

$$\begin{aligned} f_X(\boldsymbol{x}_{\mathrm{Uk}}\,|\boldsymbol{x}_{\mathrm{Kn}}) &= \frac{f_X(\boldsymbol{x}_{\mathrm{Kn}}, \boldsymbol{x}_{\mathrm{Uk}})}{f_X(\boldsymbol{x}_{\mathrm{Kn}})} \\ &= \frac{f_X(\boldsymbol{x} = \boldsymbol{K}\,\boldsymbol{x}_{\mathrm{Kn}} + U\boldsymbol{x}_{\mathrm{Uk}})}{f_X(\boldsymbol{x}_{\mathrm{Kn}})} \end{aligned} \tag{11.48}$$

In Equation (11.48), for a given sequence of samples $\boldsymbol{x}_{\mathrm{Kn}}$, $f_X(\boldsymbol{x}_{\mathrm{Kn}})$ is a constant. Therefore the estimate that maximises the posterior pdf, i.e. the MAP estimate, is given by

$$\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{MAP} = \arg\max_{\boldsymbol{x}_{\mathrm{Uk}}} f_X(\boldsymbol{K}\,\boldsymbol{x}_{\mathrm{Kn}} + \boldsymbol{U}\,\boldsymbol{x}_{\mathrm{Uk}}) \tag{11.49}$$

Example 11.2 MAP interpolation of a Gaussian signal

Assume that an observation signal $\boldsymbol{x} = \boldsymbol{K}\boldsymbol{x}_{\mathrm{Kn}} + \boldsymbol{U}\boldsymbol{x}_{\mathrm{Uk}}$, from a zero-mean Gaussian process, is composed of a sequence of M missing samples $\boldsymbol{x}_{\mathrm{Uk}}$ and $N - M$ known neighbouring samples as in Equation (11.47). The pdf of the signal $\boldsymbol{x}$ is given by

$$f_X(\boldsymbol{x}) = \frac{1}{(2\pi)^{N/2}\,|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{x}\right) \tag{11.50}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ is the covariance matrix of the Gaussian vector process $\boldsymbol{x}$. Substitution of Equation (11.50) in Equation (11.48) yields the conditional pdf of the unknown signal $\boldsymbol{x}_{\mathrm{Uk}}$ given a number of samples $\boldsymbol{x}_{\mathrm{Kn}}$:

$$\begin{aligned} f_X(\boldsymbol{x}_{Uk}\,|\boldsymbol{x}_{Kn}) = {} & \frac{1}{f_X(\boldsymbol{x}_{\mathrm{Kn}})}\frac{1}{(2\pi)^{N/2}\,|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \times \\ & \exp\left(-\frac{1}{2}(\boldsymbol{K}\,\boldsymbol{x}_{\mathrm{Kn}} + \boldsymbol{U}\,\boldsymbol{x}_{\mathrm{Uk}})^T\,\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\,(\boldsymbol{K}\,\boldsymbol{x}_{\mathrm{Kn}} + \boldsymbol{U}\,\boldsymbol{x}_{\mathrm{Uk}})\right) \end{aligned} \tag{11.51}$$

The MAP signal estimate, obtained by setting the derivative of the log-likelihood function ln $f_X(\boldsymbol{x}|\boldsymbol{x}_{\mathrm{Kn}})$ of Equation (11.51) with respect to $\boldsymbol{x}_{\mathrm{Uk}}$ to zero, is given by

$$\boldsymbol{x}_{\mathrm{Uk}} = -\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{K}\,\boldsymbol{x}_{\mathrm{Kn}} \tag{11.52}$$

An example of MAP interpolation is shown in Figure 11.10.

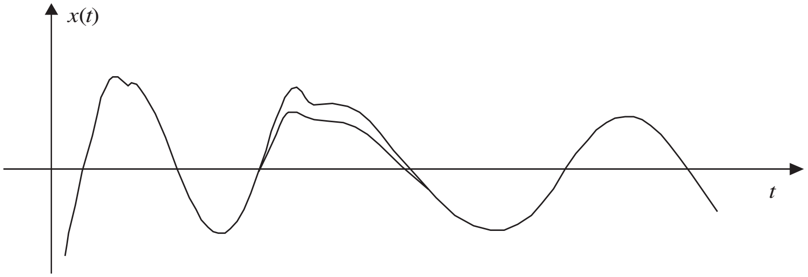


Figure 11.10 Illustration of MAP interpolation of a segment of 20 samples.

11.3.2 Least Square Error Autoregressive Interpolation

In this section, we describe interpolation based on an autoregressive (AR) 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 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 11.3.4, the AR interpolation method is extended to include pitch–period correlations.

11.3.3 Interpolation Based on a Short-Term Prediction Model

An autoregressive (AR), or linear predictive, signal $x(m)$ is described as

$$x(m)=\sum_{k=1}^{P} a_k\, x(m-k)+e(m) \tag{11.53}$$

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 quasi-periodic 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, $\boldsymbol{x}_{\mathrm{Uk}}=\{x(k),\ \ldots,\ x(k+M-1)\}$ are missing. The objective is to estimate the missing samples $\boldsymbol{x}_{\mathrm{Uk}}$, using the remaining $N-M$ samples and an AR model of the signal. Figure 11.8 illustrates the interpolation problem. For this signal record of N samples, the AR Equation (11.53) can be expanded to form the following matrix equation:

$$\begin{pmatrix} e(P)\\ e(P+1)\\ \vdots\\ e(k-1)\\ \hdashline e(k)\\ e(k+1)\\ e(k+2)\\ \vdots\\ e(k+M+P-2)\\ e(k+M+P-1)\\ \hdashline e(k+M+P)\\ e(k+M+P+1)\\ \vdots\\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(P)\\ x(P+1)\\ \vdots\\ x(k-1)\\ \hdashline x_{\mathrm{Uk}}(k)\\ x_{\mathrm{Uk}}(k+1)\\ x_{\mathrm{Uk}}(k+2)\\ \vdots\\ x(k+M+P-2)\\ x(k+M+P-1)\\ \hdashline x(k+M+P)\\ x(k+M+P+1)\\ \vdots\\ x(N-1) \end{pmatrix}$$

$$-\begin{pmatrix} x(P-1) & x(P-2) & \cdots & x(0)\\ x(P) & x(P-1) & \cdots & x(1)\\ \vdots & \vdots & \ddots & \vdots\\ x(k-2) & x(k-3) & \cdots & x(k-P-1)\\ \hdashline x(k-1) & x(k-2) & \cdots & x(k-P)\\ x_{Uk}(k) & x(k-1) & \cdots & x(k-P+1)\\ x_{Uk}(k+1) & x_{Uk}(k) & \cdots & x(k-P+2)\\ \vdots & \vdots & \ddots & \vdots\\ x(k+M+P-3) & x(k+M+P-4) & \cdots & x_{\mathrm{Uk}}(k+M-2)\\ x(k+M+P-2) & x(k+M+P-3) & \cdots & x_{\mathrm{Uk}}(k+M-1)\\ \hdashline x(k+M+P-1) & x(k+M+P-2) & \cdots & x(k+M)\\ x(k+M+P) & x(k+M+P-1) & \cdots & x(k+M+1)\\ \cdots & \cdots & \ddots & \cdots\\ x(N-2) & x(N-3) & \cdots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1\\ a_2\\ a_3\\ \vdots\\ a_P \end{pmatrix} \tag{11.54}$$

where the subscript Uk denotes the unknown samples. Equation (11.54) can be rewritten in compact vector notation as

$$\boldsymbol{e}\left(\boldsymbol{x}_{\mathrm{UK}},\boldsymbol{a}\right)=\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a} \tag{11.55}$$

where the error vector $\boldsymbol{e}(\boldsymbol{x}_{\mathrm{Uk}},\ \boldsymbol{a})$ 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 $\boldsymbol{a}$ and the missing samples $\boldsymbol{x}_{\mathrm{Uk}}$ is the minimum mean square error given by the inner vector product

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}\left(\boldsymbol{x}_{\mathrm{UK}},\boldsymbol{a}\right)=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{a}-2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x} \tag{11.56}$$

The squared error function in Equation (11.56) involves nonlinear unknown terms of fourth order, $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{a}$, and cubic order, $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}$. The least square error formulation, obtained by differentiating $\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{x}_{\mathrm{Uk}},\ \boldsymbol{a})$, with respect to the vectors $\boldsymbol{a}$ or $\boldsymbol{x}_{\mathrm{Uk}}$, results in a set of nonlinear equations of cubic order whose solution is non-trivial. 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-and-maximise (EM) algorithm, and is similar to the linear-predictive model-based restoration considered in Section 8.7. In the first stage of the solution, Equation (11.54) is *linearised* by either assuming that the missing samples have zero values or discarding the set of equations in (11.54), between the two dashed lines, that involve the unknown signal samples. The linearised equations are used to solve for the AR model coefficient vector $\boldsymbol{a}$ by forming the equation

$$\hat{\boldsymbol{a}}=\left(\boldsymbol{X}_{\mathrm{Kn}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{Kn}}\right)^{-1}\left(\boldsymbol{X}_{\mathrm{Kn}}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{Kn}}\right) \tag{11.57}$$

where the vector $\hat{\boldsymbol{a}}$ 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 $\boldsymbol{x}_{\mathrm{Uk}}$. For an AR model of order P, and an unknown signal segment of length M, there are $2M+P$ nonlinear equations in (11.54) that involve the unknown samples; these are

$$\begin{pmatrix} e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \end{pmatrix} = \begin{pmatrix} x_{\mathrm{Uk}}(k) \\ x_{\mathrm{Uk}}(k+1) \\ x_{\mathrm{Uk}}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \end{pmatrix}$$
$$-\begin{pmatrix} x(k-1) & x(k-2) & \dots & x(k-p) \\ x_{\mathrm{Uk}}(k) & x(k-1) & \dots & x(k-p+1) \\ x_{\mathrm{Uk}}(k+1) & x_{\mathrm{Uk}}(k) & \dots & x(k-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(k+M+P-3) & x_{\mathrm{Uk}}(k+M+P-4) & \dots & x_{\mathrm{Uk}}(k+M-2) \\ x(k+M+P-2) & x_{\mathrm{Uk}}(k+M+P-3) & \dots & x_{\mathrm{Uk}}(k+M-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{P-1} \\ a_P \end{pmatrix} \tag{11.58}$$

The estimate of the predictor coefficient vector, a, obtained from the first stage of the solution, is substituted in Equation (11.58) so that the only remaining unknowns in (11.58) are the missing signal samples. Equation (11.58) may be partitioned and rearranged in vector notation in the following form:

$$\begin{aligned}
\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 & \cdots & 0 \\
-a_1 & 1 & 0 & 0 & \cdots & 0 \\
-a_2 & -a_1 & 1 & 0 & \cdots & 0 \\
-a_3 & -a_2 & -a_1 & 1 & \cdots & 0 \\
-a_4 & -a_3 & -a_2 & -a_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
-a_P & -a_{P-1} & -a_{P-2} & -a_{P-3} & \cdots & 0 \\
0 & -a_P & -a_{P-1} & -a_{P-2} & \cdots & 0 \\
0 & 0 & -a_P & -a_{P-1} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \cdots & -a_{P-1} \\
0 & 0 & 0 & 0 & \cdots & -a_P
\end{pmatrix}
\begin{pmatrix} x_{\mathrm{Uk}}(k) \\ x_{\mathrm{Uk}}(k+1) \\ x_{\mathrm{Uk}}(k+2) \\ x_{\mathrm{Uk}}(k+3) \\ \vdots \\ x_{\mathrm{Uk}}(k+M-1) \end{pmatrix} + \\
&\begin{pmatrix}
-a_P & -a_{P-1} & -a_{P-2} & \cdots & -a_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & -a_P & -a_{P-1} & \cdots & -a_2 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & -a_P & \cdots & -a_3 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -a_P & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_2 & -a_1 & 1 & \cdots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_3 & -a_2 & -a_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & 0 \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_{P-1} & -a_{P-2} & -a_{P-3} & \cdots & 1
\end{pmatrix}
\begin{pmatrix} x(k-P) \\ x(k-P+1) \\ x(k-P+2) \\ \vdots \\ x(k-1) \\ 0 \\ \vdots \\ x(k+M) \\ x(k+M+1) \\ x(k+M+2) \\ \vdots \\ x(k+M+P-1) \end{pmatrix}
\end{aligned} \tag{11.59}$$

In Equation (11.59), the unknown and known samples are rearranged and grouped into two separate vectors. In a compact vector–matrix notation, Equation (11.58) can be written in the form

$$\boldsymbol{e} = \boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}} + \boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}} \tag{11.60}$$

where $\boldsymbol{e}$ is the error vector, $\boldsymbol{A}_1$ is the first coefficient matrix, $\boldsymbol{x}_{\mathrm{Uk}}$ is the unknown signal vector being estimated, $\boldsymbol{A}_2$ is the second coefficient matrix and the vector $\boldsymbol{x}_{\mathrm{Kn}}$ consists of the *known* samples in the signal matrix and vectors of Equation (11.58). The total squared error is given by

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = (\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}} + \boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}})^{\mathrm{T}}(\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}} + \boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}}) \tag{11.61}$$

The least square AR (LSAR) interpolation is obtained by minimisation of the squared error function with respect to the unknown signal samples $\boldsymbol{x}_{\mathrm{Uk}}$:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{x}_{\mathrm{Uk}}} = 2\boldsymbol{A}_1{}^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}} + 2\boldsymbol{A}_1{}^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}} = 0 \tag{11.62}$$

From Equation (11.62) we have

$$\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{LSAR} = -\left(\boldsymbol{A}_1{}^{\mathrm{T}}\boldsymbol{A}_1\right)^{-1}\left(\boldsymbol{A}_1{}^{\mathrm{T}}\boldsymbol{A}_2\right)\boldsymbol{x}_{\mathrm{Kn}} \tag{11.63}$$

The solution in Equation (11.62) gives the $\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{LSAR}$, vector which is the least square error estimate of the unknown data vector.

11.3.4 Interpolation Based on Long-Term 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 quasi-periodic 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 quasi-periodic excitation pulse.

In this section, the AR interpolation is extended to include both long-term 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 be usually modelled with an AR model of order 10–20. In order to include the pitch periodicities in the AR model of Equation (11.53), the model order must be greater than the pitch period. For speech signals, the pitch period is normally in the range 4–20 milliseconds, 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 11.11. This modified AR model may be expressed by the following equation:

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + \sum_{k=-Q}^{Q} p_k x(m-T-k) + e(m) \tag{11.64}$$

The AR model of Equation (11.64) is composed of a *short-term predictor* $\boldsymbol{\Sigma}_{a_k}\, x(m-k)$ that models the contribution of the P immediate past samples, and a *long-term predictor* $\boldsymbol{\Sigma}_{p_k}\, x(m-T-k)$ 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 (11.64) is specified by the parameter vector $\boldsymbol{c} = [a_1, \ldots, a_P, p_{-Q}, \ldots, p_Q]$ and the pitch period T. Note that in Figure 11.11 the sample marked '◇**?**' coincides with the onset of an excitation pulse. This sample is not well predictable from the P past samples, because they do not include a similar pulse event. The sample is more predictable from the $2Q+1$ samples a pitch period away, since they include the effects of a similar excitation pulse. In contrast, the sample marked '○**?**' is more predicable from the immediate past P sample and $2Q+1$ samples a pitch period away.

$$\boldsymbol{c} = \boldsymbol{R}_{xx}^{-1} \boldsymbol{r}_{xx} \tag{11.65}$$

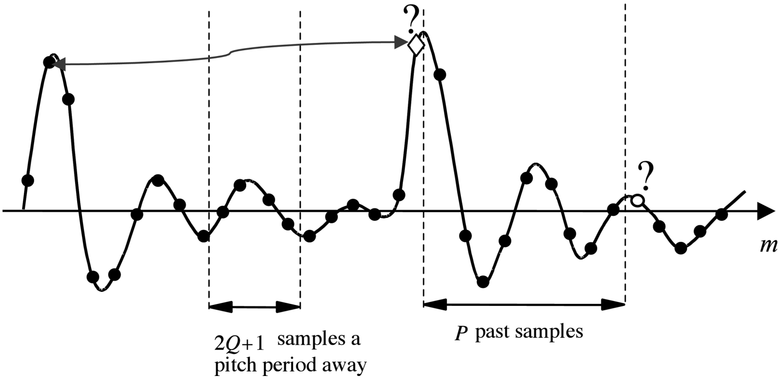


Figure 11.11 A quasiperiodic waveform. The sample marked '○?' can be predicted using P immediate past samples and $2Q+1$ samples a pitch period away.

where $\boldsymbol{R}_{xx}$ is the autocorrelation matrix of signal $\boldsymbol{x}$ and $\boldsymbol{r}_{\mathbf{xx}}$ is the correlation vector. In expanded form, Equation (11.65) can be written as

$$
\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{+Q} \end{pmatrix}
=
\begin{pmatrix}
r(0) & r(1) & \cdots & r(P-1) & r(T+Q-1) & r(T+Q) & \cdots & r(T-Q-1) \\
r(1) & r(0) & \cdots & r(P-2) & r(T+Q-2) & r(T+Q-1) & \cdots & r(T+Q-2) \\
r(2) & r(1) & \cdots & r(P-3) & r(T+Q-3) & r(T+Q-2) & \cdots & r(T+Q-3) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
r(P-1) & r(P-2) & \cdots & r(0) & r(T+Q-P) & r(T+Q-P+1) & \cdots & r(T+Q-P) \\
r(T+Q-1) & r(T+Q-2) & \cdots & r(T+Q-P) & r(0) & r(1) & \cdots & r(2Q) \\
r(T+Q) & r(T+Q-1) & \cdots & r(T+Q-P+1) & r(1) & r(0) & \cdots & r(2Q-1) \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
r(T-Q-1) & r(T-Q-2) & \cdots & r(T-Q-P) & r(2Q) & r(2Q-1) & \cdots & r(0)
\end{pmatrix}^{-1}
\begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T+Q) \\ r(T+Q-1) \\ \vdots \\ r(T-Q) \end{pmatrix}
\tag{11.66}
$$

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, $\boldsymbol{x}_{\mathrm{Uk}} = \{x(k),\ x(k+1), \ldots, x(k+M-1)\}$ is missing. Figure 11.12 illustrates the interpolation problem. The missing samples are estimated using P samples in the immediate vicinity and $2Q+1$ samples a pitch period away on each side of the missing signal. For the signal record of N samples, the modified AR Equation (11.64) can be written in matrix form as

$$
\left(\begin{array}{c}
e(T+Q) \\ e(T+Q+1) \\ \vdots \\ e(k-1) \\ \hdashline
e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \\ \hdashline
e(k+M+P) \\ e(k+M+P+1) \\ \vdots \\ e(N-1)
\end{array}\right)
=
\left(\begin{array}{c}
x(T+Q) \\ x(T+Q+1) \\ \vdots \\ x(k-1) \\ \hdashline
x_{UK}(k) \\ x_{UK}(k+1) \\ x_{UK}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \\ \hdashline
x(k+M+P) \\ x(k+M+P+1) \\ \vdots \\ x(N-1)
\end{array}\right)
$$

$$
-\left(\begin{array}{ccccc}
x(T+Q-1) & \cdots & x(T+Q-P) & x(2Q) & \cdots & x(0) \\
x(T+Q) & \cdots & x(T+Q-P+1) & x(2Q+1) & \cdots & x(1) \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
x(k-2) & \cdots & x(k-P-1) & x(k-T+Q-1) & \cdots & x(k-T-Q-1) \\ \hdashline
x(k-1) & \cdots & x(k-P) & x(k-T+Q) & \cdots & x(k-T-Q) \\
x_{UK}(k) & \cdots & x(k-P+1) & x(k-T+Q+1) & \cdots & x(k-T-Q+1) \\
x_{UK}(k+1) & \cdots & x(k-P+2) & x(k-T+Q+2) & \cdots & x(k-T-Q+2) \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
x(k+M+P-3) & \cdots & x_{UK}(k+M-2) & x(k+M+P-T+Q-2) & \cdots & x(k+M+P-T-Q-2) \\
x(k+M+P-2) & \cdots & x_{UK}(k+M-1) & x(k+M+P-T+Q-1) & \cdots & x(k+M+P-T-Q-1) \\ \hdashline
x(k+M+P-1) & \cdots & x(k+M) & x(k+M+P-T+Q) & \cdots & x(k+M+P-T-Q) \\
x(k+M+P) & \cdots & x(k+M+1) & x(k+M+P-T+Q+1) & \cdots & x(k+M+P-T-Q+1) \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
x(N-2) & \cdots & x(N-P-1) & x(N-T+Q-1) & \cdots & x(N-T-Q-1)
\end{array}\right)
\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ \vdots \\ p_{+Q} \end{pmatrix}
\tag{11.67}
$$

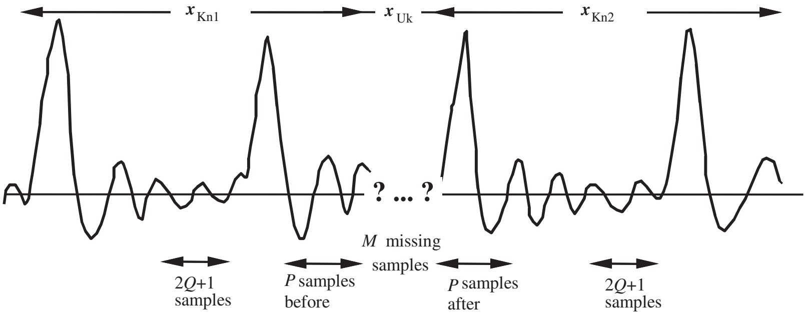


Figure 11.12 A signal with M missing samples. P immediate samples each side of the gap and $2Q+1$ samples a pitch period away are used for interpolation.

where the subscript Uk denotes the unknown samples. In compact matrix notation, this set of equation can be written in the form

$$\boldsymbol{e}\,(\boldsymbol{x}_{Uk},\boldsymbol{c})=\boldsymbol{x}+\boldsymbol{X}\boldsymbol{c} \tag{11.68}$$

As in Section 11.3.2, 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 $\boldsymbol{c}$.
(2) In the second stage, the AR coefficient estimates are substituted in Equation (11.68) so that the only unknowns are the data samples.

The solution follows the same steps as those described in Section 11.3.2.

11.3.5 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

$$v(m)=x(m)-\hat{x}(m) \tag{11.69}$$

A common measure of signal distortion is the mean square error distance defined as

$$D(\mathbf{c},M)=\frac{1}{M}\mathcal{E}\left\{\sum_{m=0}^{M-1}\left[x(k+m)-\hat{x}(k+m)\right]^2\right\} \tag{11.70}$$

where k is the beginning of an M-samples long segment of missing signal, and $\mathcal{E}$ [.] is the expectation operator. In Equation (11.70), the average distortion D is expressed as a function of the number of the unknown samples M, and also the model coefficient vector $\boldsymbol{c}$. 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, increase 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 (11.63) 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 quasi-periodic 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 2–3 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 (11.54), 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 $\boldsymbol{x}_{\mathrm{Uk}} = \mathbf{0}$), Equation (11.54) 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 11.13 and 11.14 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 (11.53) and also by the extended AR model of Equation (11.64). The length of the conventional linear predictor model was set to 20. The modified linear AR model of Equation (11.64) has a prediction order of (20,7); that is, the short-term predictor has 20 coefficients and the long-term predictor has 7 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.

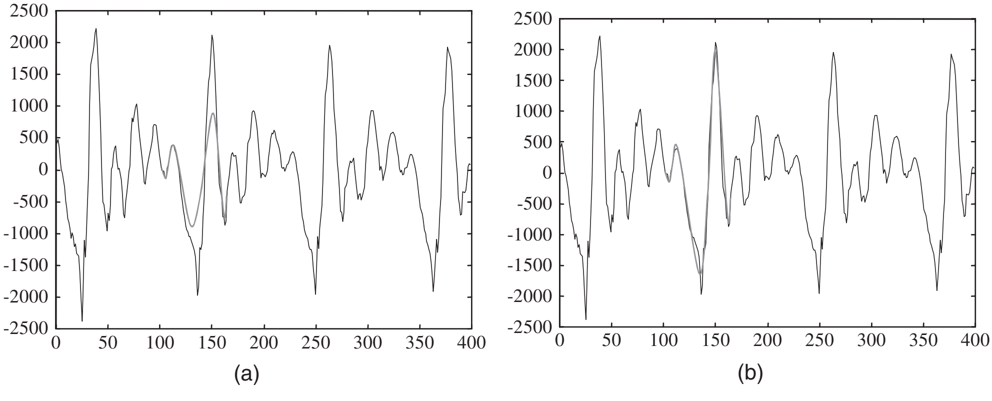


Figure 11.13 (a) A section of speech showing interpolation of 60 samples starting from the sample point 100 (b) Interpolation using short and long-term correlations. Interpolated samples are shown by the light shaded line.

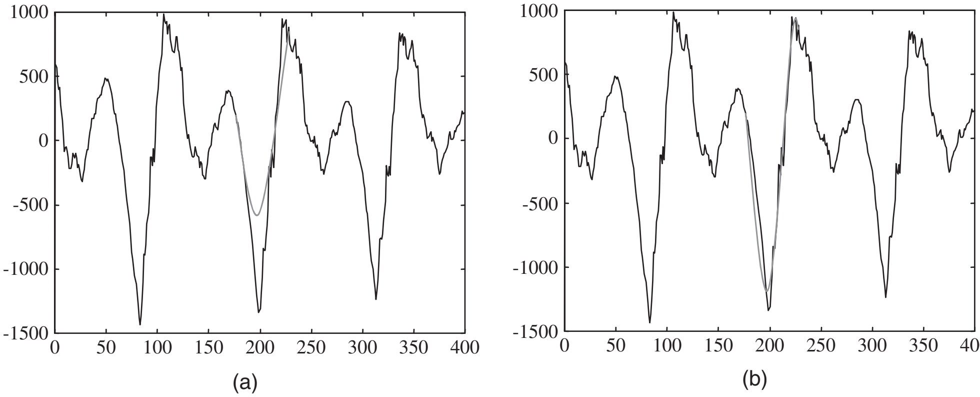


Figure 11.14 (a) A section of speech showing interpolation of 50 samples starting from the sample point 175 (b) Interpolation using short and long-term correlations. Interpolated samples are shown by the light shaded line.

11.3.6 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 (11.58).

Spectral–time representation provides a useful form for the interpolation of a large gap of missing samples. For example, through discrete Fourier transformation (DFT) 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 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 11.15. 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

$$X(k,m)=\sum_{i=0}^{N-1} w(i)x\left(m(N-D)+i\right)e^{-j\frac{2\pi}{N}ik}, \quad k=0,\ldots,N-1 \tag{11.71}$$

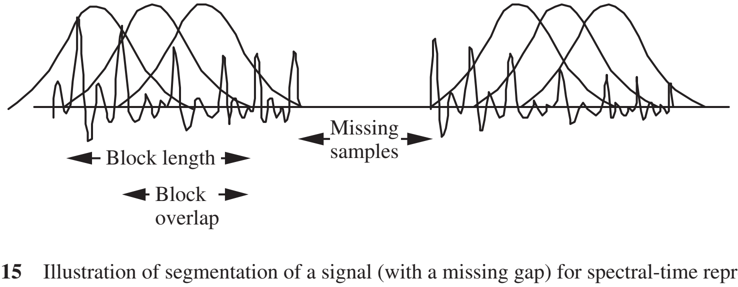


Figure 11.15 Illustration of segmentation of a signal (with a missing gap) for spectral-time representation.

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 different characteristics. Figure 11.16 illustrates a spectral-time signal with a missing block of samples. The aim of 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.

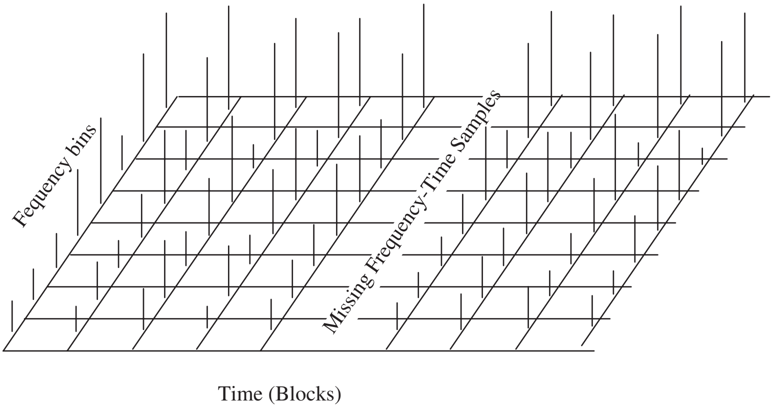


Figure 11.16 Spectral-time representation of a signal with a missing gap.

11.3.7 Interpolation Using Adaptive Code Books

In the LSAR interpolation method, described in Section 11.3.2, 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 11.17.

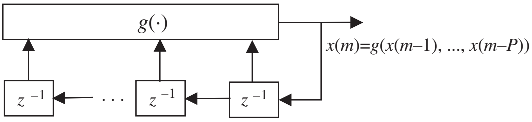


Figure 11.17 Configuration of a digital oscillator.

The general form of the equation describing a digital nonlinear oscillator can be expressed as

$$x(m) = g_f\,(x(m-1), x(m-2), \ldots, x(m-P)) \tag{11.72}$$

The mapping function $g_f(\cdot)$ may be a parametric or a non-parametric mapping. The model in Equation (11.72) 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 non-parametric method for its ease of formulation and stable characteristics. Kubin and Kleijin (1994) have described a non-parametric 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 $(P+1)^{\text{th}}$ sample is intended as an output. Given P input samples $\boldsymbol{x} = [x(m-1),\ \ldots,\ x(m-P)]$, the codebook output is the $(P+1)^{\text{th}}$ sample of the vector in the codebook whose first P samples have a minimum distance from the input signal $\boldsymbol{x}$. 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

$$x_b(m) = g_b\big(x(m+1), x(m+2), \cdots, x(m+P)\big) \tag{11.73}$$

As in the case of a forward oscillator, the backward oscillator can be designed using a non-parametric method based on an adaptive codebook of the signal process. In this case each entry in the code book has $P+1$ samples where the first sample is intended as an output sample. Given P input samples $\boldsymbol{x} = [x(m),\ \ldots,\ x(m+P-1)]$ the codebook output is the first sample of the code book vector whose next P samples have a minimum distance from the input signal $\boldsymbol{x}$.

For interpolation of M missing samples, the outputs of the forward and backward nonlinear oscillators may be combined as

$$\hat{x}(k+m) = \left(\frac{M-1-m}{M-1}\right)\hat{x}_f(k+m) + \left(\frac{m}{M-1}\right)\hat{x}_b(k+m) \tag{11.74}$$

where it is assumed that the missing samples start at k.

11.3.8 Interpolation Through Signal Substitution

Audio signals often have a time-varying but quasi-periodic 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 (1989).

11.3.9 LP-HNM Model based Interpolation

In section 17.4 a high performance interpolation method is described, for the replacement of lost speech segments, using a combination of a linear prediction (LP) model of the spectral envelop of speech and a harmonic plus noise model (HNM) of excitation. The LP and HNM parameters are extracted and interpolated separately and the interpolated models are combined to reconstitute the missing samples. The method is described in detail in section 17.4.

11.4 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-analog 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 code book 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 to 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. *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, **ICASSP-89**: 180–183, Glasgow.

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, March.

Godsill S.J. (1993) The Restoration of Degraded Audio Signals. Ph.D. Thesis, Cambridge University.

Godsill S.J. and Rayner P.J.W. (1993) Frequency domain interpolation of sampled signals. *IEEE Int. Conf., Speech and Signal Processing*, **ICASSP-93**, Minneapolis.

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. Acoustics, Speech and Signal Processing*, **ASSP-34**(2): 317–330 June.

Janssen A.J. and Vries L.B. (1984) Interpolation of Band-Limited Discrete-Time Signals by Minimising Out-of Band Energy. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, **ICASSP-84**.

Kay S.M. (1983) Some Results in Linear Interpolation Theory. *IEEE Trans. Acoustics Speech and Signal Processing*, **ASSP-31**: 746–749, June.

Kay S.M. (1988) *Modern Spectral Estimation: Theory and Application*. Prentice- Hall, Englewood Cliffs, NJ.

Kolmogrov A.N. (1939) Sur l' Interpolation et Extrapolation des Suites Stationaires, *Comptes Rendus de l'Academie des Sciences*. **208**: 2043–2045.

Kubin G. and Kleijin W.B. (1994) Time-Scale Modification of Speech Based On A Nonlinear Oscillator Model. *Proc. IEEE Int. Conf., Speech and Signal Processing*, **ICASSP-94**: I453–I456, Adelaide.

Lochart G.B. and Goodman D.J. (1986) Reconstruction of Missing Speech Packets by Waveform Substitution. Signal Processing 3: *Theories and Applications*: 357–360.

Marks R.J. (1983) Restoring Lost Samples From An Over-Sampled Band-Limited Signal. *IEEE Trans. Acoustics, Speech and Signal Processing*, **ASSP-31**(2):752–755, June.

Marks R.J. (1991) *Introduction to Shannon Sampling and Interpolation Theory*. Springer Verlag.

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, MA.

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 Vettereling W.T. (1992) *Numerical Recepies in C*, 2nd edn. Cambridge University Press.

Nakamura S. (1991) *Applied Numerical Methods with Software*. Prentice-Hall, Englewood Cliffs, NJ.

Schafer, R.W. and Rabiner, L.R. (1973) A Digital Signal Processing Approach to Interpolation. *Proc. IEEE*, **61**: 692–702, June.

Steele R. and Jayant N.S. (1980) Statistical Block Coding for DPCM-AQF Speech. *IEEE Trans. Communications*. **COM-28**(11): 1899–1907, Nov.

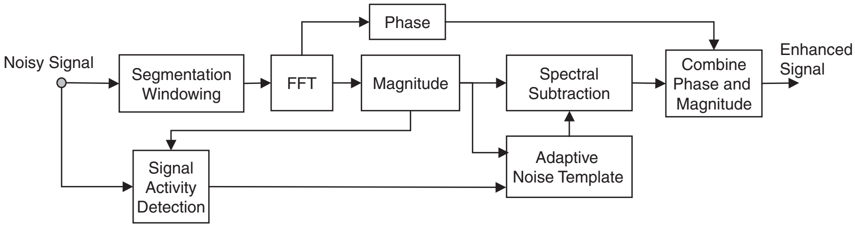
Tong H. (1990) *Nonlinear Time Series A Dynamical System Approach*. Oxford University Press.

Vaseghi S.V. (1988) Algorithms for Restoration of Gramophone Records. Ph.D. Thesis, Cambridge University.

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. *Proc. IEEE* Int. *Conf. on Acoustics, Speech and Signal Processing*, **ICASSP-93**: II-554-II-557, Adelaide.

Wiener N. (1949) *Extrapolation, Interpolation and Smoothing of Stationary Time Series With Engineering Applications*. MIT Press, Cambridge, MA.



12

Signal Enhancement via Spectral Amplitude Estimation

Spectral amplitude estimation forms the basis of most signal restoration systems, such as for speech de-noising, where the phase distortion can be ignored. Note that in severely noisy conditions, as the signal to noise ratio drops below 0 dB, phase distortion cannot 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 non-linear 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 (MMSE) cost function, as this method has received much attention in the quest for development of improved signal restoration algorithms. Finally, the critical issue of estimation of signal to noise ratios is considered.

12.1 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 1.3.4 takes as the inputs the noise and the noisy signal, and 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

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

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 12.1. The increase in the variance of the signal spectrum results from the random fluctuations of the noise.

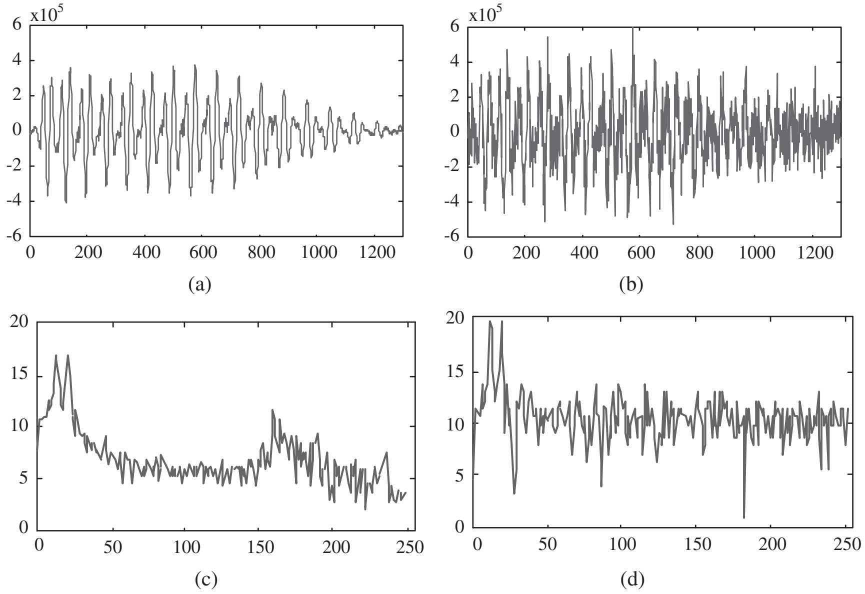


Figure 12.1 Illustrations of the effect of noise on a signal in the time and the frequency domains: (a) clean signal, (b) noisy signal, (c) spectrum of clean signal and (d) spectrum of noisy signal.

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.

12.1.1 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

$$y(m) = x(m) + n(m) \tag{12.1}$$

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 (12.1) can be represented as

$$Y(k) = X(k) + N(k) \qquad k = 0, \ldots, N-1 \tag{12.2}$$

where the complex variables $X(k)$, $N(k)$ and $Y(k)$ are the short time discrete Fourier transforms (DFT) 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/sec or kF_s/N Hz where F_s is the sampling frequency in units of Hz.

Equation (12.2), 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

$$Y_k e^{j\theta_{Y_k}} = X_k e^{j\theta_{X_k}} + N_k e^{j\theta_{N_k}} \quad k = 0, \ldots, N-1 \tag{12.3}$$

where $Y_k = |Y(k)|$ and $\theta_{Y_k} = \tan^{-1}\Big(\mathrm{Im}\big(Y(k)\big)/\mathrm{Re}\big(Y(k)\big)\Big)$ 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.

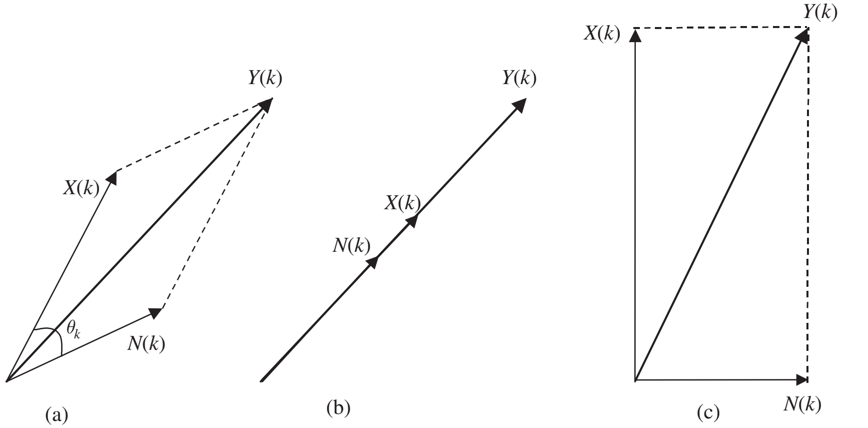
12.1.2 Vector Representation of Spectrum of Noisy Signals

The frequency 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

$$Y_k^2 = X_k^2 + N_k^2 + 2X_k N_k \cos(\theta) \qquad k = 0, \ldots, N-1 \tag{12.4}$$

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 (12.3) is modelled in the spectral amplitude estimation process, it contributes to an estimation error.

Figure (12.2.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 (12.2.b) shows that the assumption that amplitude spectrum



(a)

Spectral values of signal,
Noise and noisy signal

(b)

Assuming θ_k=0 leads to

$Y_k = X_k + N_k$

(c)

Assuming θ_k=90 leads to

$Y_k^2 = X_k^2 + N_k^2$

Figure 12.2 (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 $\pi/2$.

of the noisy signal is the sum of the amplitude spectra of signal and noise, $Y_k = X_k + N_k$, 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 (12.2.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, $Y_k^2 = X_k^2 + N_k^2$, 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.

12.2 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 12.3 illustrates a block diagram configuration of the spectral subtraction method.

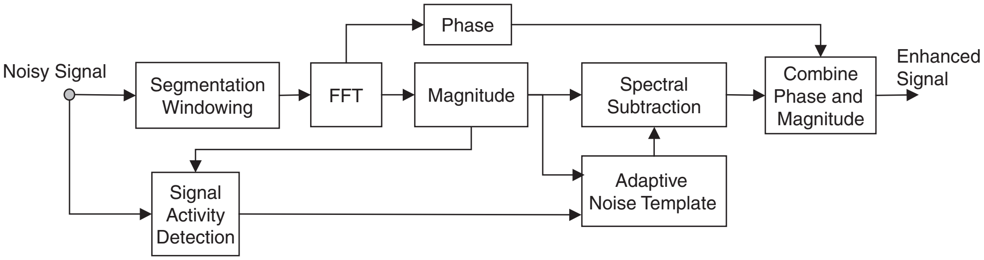


Figure 12.3 Block diagram illustration of an FFT- based spectral subtraction system for de-noising speech.

In spectral subtraction, the incoming signal $x(m)$ is divided into frames of N samples length. Each frame is windowed, using a window (i.e. Hann) and then transformed via discrete Fourier transform (DFT) to N spectral samples. The windows alleviate the effects of the discontinuities at the endpoints of each segment. The windowed signal is given by

$$\begin{aligned} y_w(m) &= w(m)y(m) \\ &= w(m)[x(m) + n(m)] \\ &= x_w(m) + n_w(m) \end{aligned} \tag{12.5}$$

The windowing operation can be expressed in the frequency domain as

$$\begin{aligned} Y_w(k) &= W(k) * Y(k) \\ &= X_w(k) + N_w(k) \end{aligned} \tag{12.6}$$

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

$$\hat{X}_k^b = Y_k^b - \alpha(k)\overline{N_k^b} \tag{12.7}$$

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 (12.7) 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

$$\overline{N_k^b} = \frac{1}{M} \sum_{i=0}^{M-1} N_{k,i}^b \tag{12.8}$$

In Equation (12.8), $N_{k,i}$ 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

$$\overline{N_{k,i}^b} = a\overline{N_{k,i-1}^b} + (1-a)N_{k,i}^b \tag{12.9}$$

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

$$\hat{x}(m) = \sum_{k=0}^{N-1} \left(\hat{X}_k e^{j\theta_{Y_k}}\right) e^{j\frac{2\pi}{N}km} \quad m = 0, \ldots, N-1 \tag{12.10}$$

where θ_{Y_k} is the phase of the noisy signal frequency $Y(k)$. The signal restoration Equation (12.10) 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 (SNR) decreases. To avoid negative magnitude estimates the spectral subtraction output is post-processed using a mapping function $T[\cdot]$ of the form

$$T[\hat{X}_k] = \begin{cases} \hat{X}_k & \text{if } \hat{X}_k > \beta Y_k \\ fn[Y_k] & \text{otherwise} \end{cases} \tag{12.11}$$

For example, we may chose a rule such that if the estimate $\hat{X}_k > 0.01\ Y_k$ (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, $fn[Y_k] = \textit{noise floor}$, where the noise floor is a positive constant. An alternative choice is $fn[Y_k] = \beta Y_k$. In this case,

$$T[\hat{X}_k] = \begin{cases} \hat{X}_k & \text{if } \hat{X}_k > \beta Y_k \\ \beta Y_k & \text{otherwise} \end{cases} \tag{12.12}$$

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.

12.2.1 Power Spectrum Subtraction

The power spectrum subtraction, or squared-magnitude spectrum subtraction, is defined by the following equation:

$$\hat{X}_k^2 = Y_k^2 - \overline{N_k^2} \tag{12.13}$$

where it is assumed that $\alpha(k)$ the subtraction factor in Equation (12.7), is one. Note that Equation (12.13) can be deduced from Equation (12.4) if we assume that the signal and noise spectral vectors are perpendicular as illustrated in Figure 12.2(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 (12.13) may be rewritten as

$$\hat{X}_k^2 = X_k^2 + \underbrace{\left(N_k^2 - \overline{N_k^2}\right)}_{\text{Noise variations}} + \underbrace{X_k^* N_k + X_k N_k^*}_{\text{Cross products}} \tag{12.14}$$

Taking the expectations of both sides of Equation (12.14), and assuming that the signal and the noise processes are uncorrelated ergodic processes, we have

$$\mathcal{E}[\hat{X}_k^2] = \mathcal{E}[X_k^2] \tag{12.15}$$

From Equation (12.15), 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 non-stationary 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 (12.15), we have not considered non-linear rectification of the negative estimates of the squared magnitude spectrum.

12.2.2 Magnitude Spectrum Subtraction

The magnitude spectrum subtraction is defined as

$$\hat{X}_k = Y_k - \overline{N_k} \tag{12.16}$$

where $\overline{N_k}$ is the time-averaged magnitude spectrum of the noise. Note that Equation (12.16) can be deduced from Equation (12.4) if we assume that the signal and noise spectral vectors are in phase as illustrated in Figure (12.2.b). Taking the expectation of Equation (12.16), we have

$$\begin{aligned}\hat{X}_k &= \mathcal{E}[Y_k] - \mathcal{E}[\overline{N_k}]\\ &= \mathcal{E}[|X(k) + N(k)|] - \mathcal{E}[\left|\overline{N_k}\right|]\\ &= \mathcal{E}[X_k]\end{aligned} \tag{12.17}$$

For signal restoration the magnitude estimate is combined with the phase of the noisy signal and then transformed into the time domain using Equation (12.10).

12.2.3 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

$$\hat{X}_k^2 = Y_k^2 - \overline{N_k^2} = H_k Y_k^2 \tag{12.18}$$

where H_k, the frequency response of the spectral subtraction filter, at discrete frequency k, is defined as

$$H_k = 1 - \frac{\overline{N_k^2}}{Y_k^2} = \frac{Y_k^2 - \overline{N_k^2}}{Y_k^2} \tag{12.19}$$

The spectral subtraction filter H_k is a zero-phase filter, with its magnitude response in the range $0 \geq H_k \geq 1$. The filter acts as a 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

$$W_k = \frac{\mathcal{E}[Y_k^2] - \mathcal{E}[N_k^2]}{\mathcal{E}[Y_k^2]} \tag{12.20}$$

A comparison of W_k and H_k, from Equations (12.20) and (12.19), 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 (12.20) with the time-averaged spectrum $\overline{Y_k^2}$, to obtain

$$H_k = \frac{\overline{Y_k^2} - \overline{N_k^2}}{\overline{Y_k^2}} \tag{12.21}$$

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 (12.21) approaches the Wiener filter Equation (12.20). In practice, many signals, such as speech and music, are non-stationary, and only a limited degree of beneficial time-averaging of the spectral parameters can be expected.

12.2.4 Processing Distortions

The main problem in spectral subtraction is the non-linear processing distortions caused by the random variations of the noise spectrum. From Equation (12.12) 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 non-linear mapping of the spectral estimates that fall below a threshold.

The same sources of distortions appear in both the magnitude and the power spectrum subtraction methods. Of the three sources of distortions listed above, the dominant distortion is often due to the non-linear mapping of the negative, or small-valued, spectral estimates. This distortion produces a metallic sounding noise, known as '*musical tone noise*' due to their 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 12.4. 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.

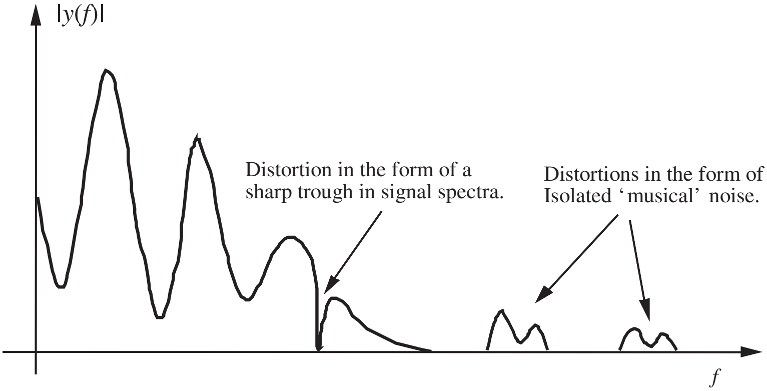


Figure 12.4 Illustration of processing distortions that may result from spectral subtraction.

12.2.5 *Effect of Spectral Subtraction on Signal Distribution*

Figure 12.5 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 12.5 is a plot of the high-frequency spectrum versus the low-frequency spectrum, in a two-dimensional signal space. Figure 12.5(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 12.5(b), is an increase in the mean and the variance of the spectrum, by an amount that depends

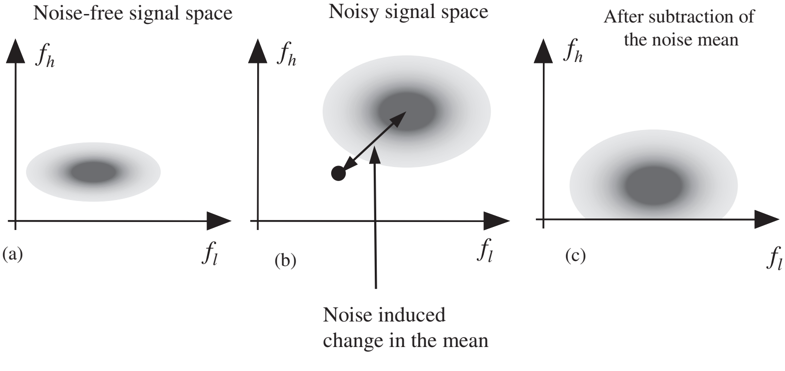


Figure 12.5 Illustration of the distorting effect of spectral subtraction on the space of the magnitude spectrum of a signal.

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 12.5(c) illustrates the distorting effect 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 those with a low SNR, become negative and have to be mapped to non-negative values. As shown this process distorts the distribution of the low-SNR part of the signal spectrum.

12.2.6 Reducing the Noise Variance

The distortions that result from spectral subtraction are due to the variations of the noise spectrum. In Section 10.4 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

$$\mathrm{Var}\left[N_k^2\right] \approx P_{NN}^2(k) = \sigma_n^4 \tag{12.22}$$

and the variance of the running average of K independent spectral components is

$$\mathrm{Var}\left[\frac{1}{K}\sum_{i=0}^{K-1} N_{k,i}^2\right] \approx \frac{1}{K}P_{NN}^2(k) \approx \frac{1}{K}\sigma_n^4 \tag{12.23}$$

From Equation (12.23), 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 non-stationary 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

$$\overline{Y_{k,i}} = \rho\overline{Y_{k,i-1}} + (1-\rho)Y_{k,i} \tag{12.24}$$

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.

12.2.7 Filtering Out the Processing Distortions

Audio signals, such as speech and music, are composed of sequences of non-stationary 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 12.6 demonstrates a 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.

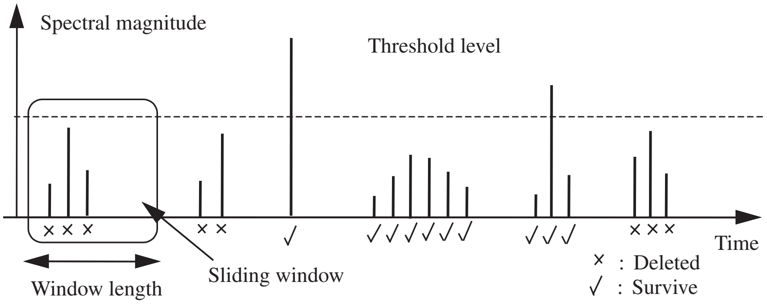


Figure 12.6 Illustration of a method for identification and filtering of 'musical noise'.

12.2.8 Non-Linear Spectral Subtraction

The use of spectral subtraction in its basic form of Equation (12.7) 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.

Non-linear spectral subtraction methods are heuristic methods that utilise estimates of the local SNR, and the observation that at a low SNR over-subtraction can produce improved results. For an explanation of the improvement that can result from over-subtraction, consider the following expression of the basic spectral subtraction equation:

$$\begin{aligned}\hat{X}_k &= Y_k - \overline{N_k} \\ &\approx X_k + N_k - \overline{N_k} \\ &\approx X_k + V_N(k)\end{aligned} \qquad (12.25)$$

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, over-subtraction, followed by non-linear 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 non-linear variants of spectral subtraction may be described by the following equation:

$$\hat{X}_k = Y_k - \alpha\big(SNR(k)\big)\overline{N_k} \tag{12.26}$$

where $\alpha(SNR(k))$ 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

$$\hat{X}_k = \begin{cases} \hat{X}_k & \text{if} \quad \hat{X}_k > \beta Y_k \\ \beta Y_k & \quad\text{otherwise} \end{cases} \tag{12.27}$$

One form of an SNR-dependent subtraction factor for Equation (12.26) is given by

$$\alpha\big(SNR(k)\big) = 1 + \frac{sd\,(N_k)}{\overline{N_k}} \tag{12.28}$$

where the function $sd(N_k)$ is the standard deviation of the noise at discrete frequency k. For white noise, $sd(N_k) = \sigma_n$, where σ_n^2 is the noise variance. Substitution of Equation (12.28) in Equation (12.26) yields

$$\hat{X}_k = Y_k - \left[1 + \frac{sd\,(N_k)}{\overline{N}_k}\right]\overline{N}_k \tag{12.29}$$

In Equation (12.29) the subtraction factor depends on the mean and the variance of the noise. Note that the amount over-subtracted 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, $\alpha\big(SNR(f)\big) = 1$, and at the other extreme for white noise $\alpha\big(SNR(f)\big) = 2$. In application of spectral subtraction to speech recognition, it is found that the best subtraction factor is usually between 1 and 2.

In the non-linear spectral subtraction method of Lockwood and Boudy, the spectral subtraction filter is obtained from

$$H_k = \frac{\overline{Y_k^2} - NL(\overline{N_k^2})}{\overline{Y_k^2}} \tag{12.30}$$

Lockwood and Boudy suggested the following function as a non-linear estimator of the noise spectrum:

$$NL\left(\overline{N_k^2}\right) = \Phi\left(\max_{\text{over } M \text{ frames}}\left(N_k^2\right), SNR(k), \overline{N_k^2}\right) \tag{12.31}$$

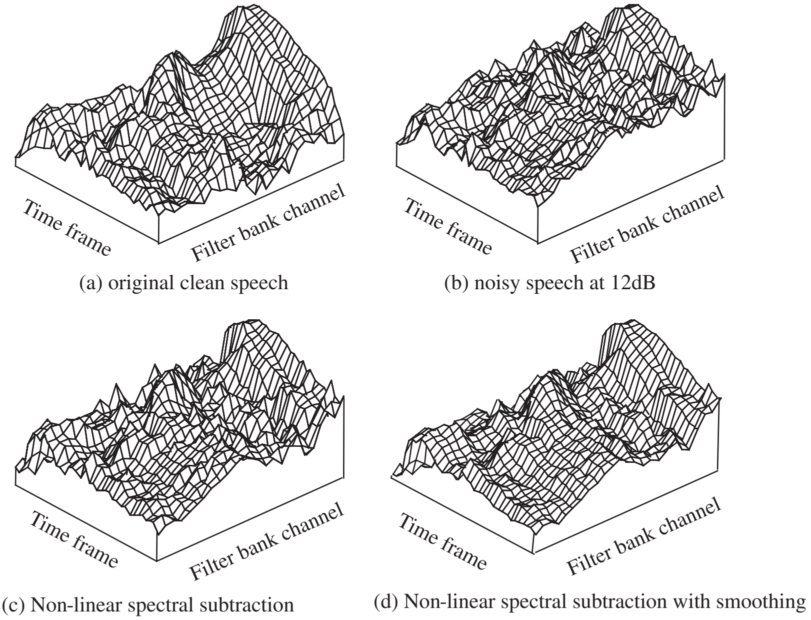
The estimate of the noise spectrum is a function of the maximum value of noise spectrum over M frames, and the signal-to-noise ratio. One form for the non-linear function $\boldsymbol{\Phi}(\cdot)$ is given by the following equation:

$$\Phi\left(\max_{\text{over } M \text{ frames}}\left(N_k^2\right), SNR(k)\right) = \frac{\displaystyle\max_{\text{Over } M \text{ frames}}\left(N_k^2\right)}{1 + \gamma SNR(k)} \tag{12.32}$$

where γ is a design parameter. From Equation (12.32) as the SNR decreases the output of the non-linear estimator $\Phi(\cdot)$ approaches $\max(N_k^2)$, and as the SNR increases it approaches zero. For over-subtraction, the noise estimate is forced to be an over-estimation by using the following limiting function:

$$\overline{N_k^2} \le \Phi\left(\max_{\text{over } M \text{ frames}}\left(N_k^2\right), SNR(k), \overline{N_k^2}\right) \le 3\overline{N_k^2} \tag{12.33}$$

The maximum attenuation of the spectral subtraction filter is limited to $H_k \ge \beta$, where usually the lower bound $\beta \ge 0.01$. Figure 12.7 illustrates the effects of non-linear spectral subtraction and smoothing in restoration of the spectrum of a speech signal.



(a) original clean speech

(b) noisy speech at 12dB

(c) Non-linear spectral subtraction

(d) Non-linear spectral subtraction with smoothing

Figure 12.7 Illustration of the effects of non-linear spectral subtraction.

12.2.9 Implementation of Spectral Subtraction

Figure 12.8 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 (DFT) for transforming the time domain signal to the frequency domain; the DFT is followed by a magnitude operator;
(3) a lowpass filter (LPF) 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 12.8. Each block is windowed, and then transformed via a DFT to the frequency domain. After 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 milliseconds is used. At a sampling rate of say 20 kHz, this translates to a value for N in the range of 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 the time, and require a larger

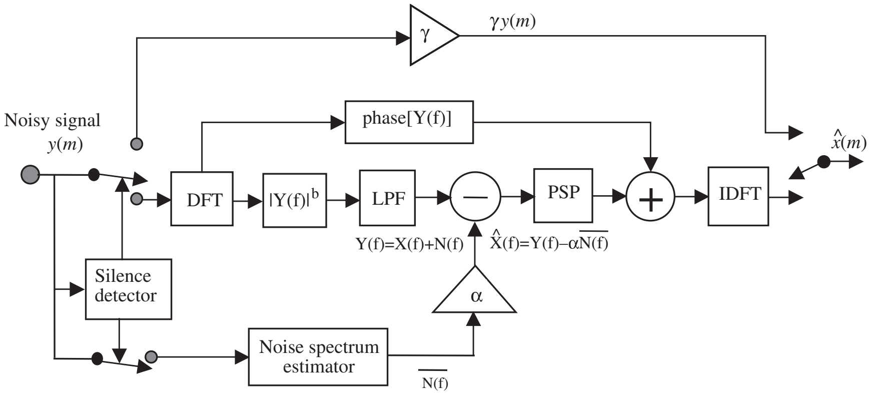


Figure 12.8 Block diagram configuration of a spectral subtraction system. PSP = post spectral subtraction processing.

window for a stable estimate. The conflicting requirement is that, owing to the non-stationary 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 12.9) 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 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.

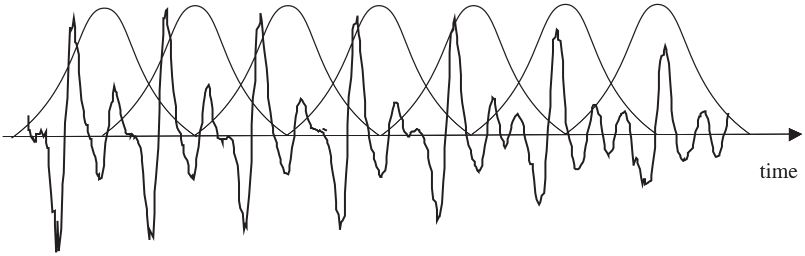


Figure 12.9 Illustration of the window and overlap process in spectral subtraction.

12.3 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 (MMSE) 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 (12.2–12.4). Assume each frame of N samples of noisy speech signal $y(m)$ is converted into N complex spectral samples $Y(k)=Y_k e^{j\theta_{Y_k}}$ via DFT. The MMSE estimation of the short-time spectral amplitude of clean speech, $\hat{X}_k$, is derived from the minimisation of a mean squared error cost function integral as

$$\hat{X}_k = \underset{\hat{X}_k}{\text{Min}}\left(\int_{-\infty}^{\infty} |X_k-\hat{X}_k|^2\, p\big(X(k)|Y(k)\big)\, dX(k)\right) \tag{12.34}$$

where $X_k-\hat{X}_k$ is the spectral amplitude estimation error $p\big(X(k)|Y(k)\big)$ and 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 (12.34) to zero, see Chapter 4, we obtain the MMSE STSA estimate as

$$\hat{X}_k = \mathcal{E}[X_k|Y(k)] = \int_{-\infty}^{\infty} X_k\, p\big(X(k)|Y(k)\big)dX(k) \quad k=0,\ldots,N-1 \tag{12.35}$$

where $\mathcal{E}[\cdot]$ is the expectation or averaging operator. *Note that the MMSE estimate of a variable is the mean or expectation of the variable.*

Using the Bayes' rule $p\big(X(k)|Y(k)\big)=p\big(Y(k)|X(k)\big)p\big(X(k)\big)/p\big(Y(k)\big)$ in Equation (12.35) we obtain

$$\hat{X}_k = \frac{\int_{-\infty}^{\infty} X_k\, p\big(Y(k)|X(k)\big)p\big(X(k)\big)dX(k)}{\int_{-\infty}^{\infty} p\big(Y(k)|X(k)\big)dX(k)} \tag{12.36}$$

Now in Equation (12.36) 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}, $p\big(X(k)\big)=p(X_k,\theta_{X_k})$, as

$$\hat{X}_k = \frac{\int_{-\infty}^{\infty}\int_{0}^{2\pi} X_k\, p\big(Y(k)|X_k,\theta_{X_k}\big)p\left(X_k,\theta_{X_k}\right) d\theta_{X_k} dX_k}{\int_{-\infty}^{\infty}\int_{0}^{2\pi} p\left(Y(k)|X_k,\theta_{X_k}\right) d\theta_{X_k} dX_k} \tag{12.37}$$

The MMSE Equation (12.37) 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

$$p\left(X_k,\theta_{X_k}\right)=p\left(\theta_{X_k}\right)p\left(X_k\right)=\frac{1}{2\pi}p\left(X_k\right) \tag{12.38}$$

Assuming that the complex spectrum of the clean signal $X(k)$ has a Gaussian distribution in the Cartesian complex domain defined as

$$p\big(X(k)\big)=\frac{1}{\pi\sigma_x^2(k)}\exp\left(-\frac{\Big(\text{Re}\big(X(k)\big)\Big)^2+\Big(\text{Im}\big(X(k)\big)\Big)^2}{2\sigma_x^2(k)}\right) \tag{12.39}$$

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

$$p(X_k)=\frac{X_k}{\sigma_x^2(k)}\exp\left(-\frac{X_k^2}{2\sigma_x^2(k)}\right) \tag{12.40}$$

where $\sigma_x^2(k)$ 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 a 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:

$$\hat{X}_k = \Gamma(1.5)\frac{\sqrt{v_k}}{\gamma_k}\exp\left(-\frac{v_k}{2}\right)\left[(1+v_k)I_0\left(\frac{v_k}{2}\right)+v_kI_1\left(\frac{v_k}{2}\right)\right]Y_k \tag{12.41}$$

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

$$v_k = \frac{\xi(k)}{1+\xi(k)}\gamma_k \tag{12.42}$$

$$\xi_k = \frac{\sigma_x^2(k)}{\sigma_n^2(k)} \tag{12.43}$$

$$\gamma_k = \frac{Y^2(k)}{\sigma_n^2(k)} \tag{12.44}$$

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 the MMSE estimator tends to the Wiener solution.

The minimum mean squared estimate of spectral amplitude can be expressed as

$$\hat{X}_k = G_{\text{MMSE}}(k)Y_k \tag{12.45}$$

where the spectral gain factor $G_{\text{MMSE}}(k)$ is given by

$$G_{\text{MMSE}}(k) = \Gamma(1.5)\frac{\sqrt{v_k}}{\gamma_k}\exp\left(-\frac{v_k}{2}\right)\left[(1+v_k)I_0\left(\frac{v_k}{2}\right)+v_kI_1\left(\frac{v_k}{2}\right)\right] \tag{12.46}$$

12.4 Estimation of Signal to Noise Ratios

One of the most critical tasks in signal enhancement methods is the accurate estimation of signal to noise ratios as a function of frequency SNR(f) $SNR(f)$. Indeed the coefficients of noise reduction filters, such as Wiener filter and spectral subtraction filter, can be expressed in terms of the SNR(f) $SNR(f)$ and furthermore the quality of signal enhancement depends on the accuracy of estimates of SNRs.

In general three types of SNRs are defined:

(1) *A priori signal to noise ratio*,
(2) *A posteriori signal to noise ratio* and
(3) *Instantaneous signal to noise*.

The *a priori signal to noise ratio spectrum* is the classical definition of signal to noise ratio expressed as the ratio of signal power (or variance) to that of noise as in Equation (12.43):

$$\xi_k = \frac{\mathcal{E}\left[|X(k)|^2\right]}{\mathcal{E}\left[|N(k)|^2\right]} = \frac{\sigma_x^2(k)}{\sigma_n^2(k)} \tag{12.47}$$

Where $\sigma_x^2(k)$ and $\sigma_n^2(k)$ are the variances of the k^{th} spectral component of signal and noise. The Wiener filter can be expressed in terms of the *a priori SNR* as $W_k = \xi_k/(1+\xi_k)$.

The *a posterior signal to noise ratio* is defined as in Equation (12.44) as the ratio of the squared of the actual noisy signal to noise power as

$$\gamma_k = \frac{|Y(k)|^2}{\mathcal{E}[|N(k)|^2]} = \frac{|Y(k)|^2}{\sigma_n^2(k)} \tag{12.48}$$

The *instantaneous signal to noise ratio* is defined as

$$\upsilon_k = \gamma_k - 1 = \frac{|Y(k)|^2 - \mathcal{E}\left[|N(k)|^2\right]}{\mathcal{E}[|N(k)|^2]} \tag{12.49}$$

Ephraim-Mallah defined a decision-directed method for estimation of the prior signal to noise ratio as

$$\hat{\xi}_k(m) = \alpha \frac{\hat{X}_k^2(m-1)}{\hat{\sigma}_n^2(k,m-1)} + (1-\alpha)P\left[\gamma(m)-1\right] \tag{12.50}$$

where $\hat{X}_k^2(m-1) = \left|\hat{X}(k,m-1)\right|^2$ is the spectral estimate for the previous frame obtained from Ephraim-Mallah's method described in the previous section and $P[\cdot]$ is a rectification function defined as

$$P[x] = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{12.51}$$

Note that in Equation (12.50) the estimate of the prior signal to noise ratio is a combination of the instantaneous SNR obtained from the spectral estimate for the previous frame $m-1$, $\hat{X}_k^2(m-1)\big/\hat{\sigma}_N^2(k,m-1)$ and estimate of the instantaneous signal to noise ratio for the current frame m obtained as $\gamma_k(m) - 1 = \left(|Y(k)|^2 - \mathcal{E}\left[|N(k)|^2\right]\right)\big/\mathcal{E}\left[|N(k)|^2\right]$.

Note that for implementation of the SNR equations the noise power is required and this may be obtained from the signal-inactive periods using a recursive smoothing formula such as

$$\hat{\sigma}_n^2(k,m) = \alpha\hat{\sigma}_n^2(k,m-1) + (1-\alpha)|N(k,m)|^2 \tag{12.52}$$

Where the lowpass filter coefficient α controls the degree of smoothing.

12.5 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 12.10 illustrates the effects of spectral subtraction in restoring a section of a speech signal contaminated with white noise. Figure 12.11 illustrates the improvement that can be obtained from application of spectral subtraction to recognition of noisy speech contaminated by a helicopter noise. The recognition results were obtained for a hidden Markov model-based spoken digit recognition.

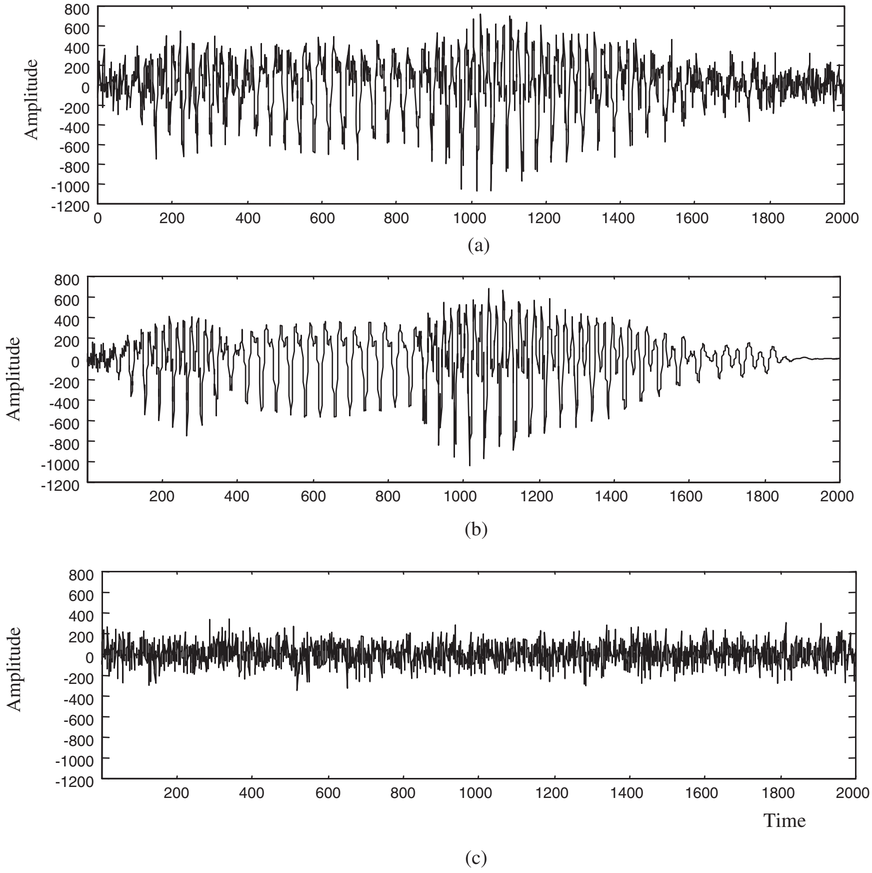


Figure 12.10 (a) A noisy signal, (b) Restored signal after spectral subtraction. (c) Noise estimate obtained by subtracting (b) from (a).

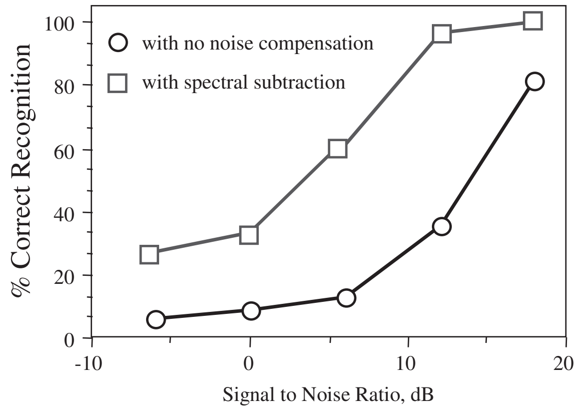


Figure 12.11 The effect of spectral subtraction in improving speech recognition (for a spoken digit data base) in the presence of helicopter noise.

12.6 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 12.2, we considered the processing distortions, and illustrated the effects of rectification of negative estimates on the distribution of the signal spectrum. In Section 12.3, a number of non-linear variants of the spectral subtraction method were considered. In signal restoration and in applications of spectral subtraction to speech recognition it is found that over-subtraction, which is subtracting more than the average noise value, can lead to improved results; if a frequency component is immersed in noise then over-subtraction can cause further attenuation of the noise. A formula is proposed in which the over-subtraction 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 Tran. Acoustics, Speech and Signal Processing* **ASSP-27**, 2: 113–120.

Brouti M., Schwartz R. and Makhoul J. (1979) Enhancement of Speech Corrupted by Acoustic Noise. *Proc. IEEE, Int. Conf. on Acoustics, Speech and Signal Processing*, **ICASSP-79**: 208–212.

Cappe O. (1994) Elimination of Musical Noise Phenomenon with the Ephraim and Malah Noise Suppressor. *IEEE Trans. Speech and Audio Processing*, **2**(2): 345–349.

Crozier P.M., Cheetham B.M.G., Holt C., Munday E. (1993) The Use of Linear Prediction and Spectral Scaling For Improving Speech Enhancement. EuroSpeech-**93**: 231–234.

Ephraim Y. (1992) Statistical Model Based Speech Enhancement systems. *Proc. IEEE*, **80**(10): 1526–1555.

Ephraim Y. and Van Trees H.L. (1993) A Signal Subspace Approach for Speech Enhancement. *Proc. IEEE, Int. Conf. on Acoustics, Speech and Signal Processing*, **ICASSP-93**: 355–358.

Ephraim Y. and Malah D. (1984) Speech Enhancement Using a Minimum Mean-Square Error Short-Time Amplitude Estimator. *IEEE Trans. Acoustics, Speech and Signal Processing*. **ASSP-32**(6): 1109–1121.

Juang B.H. and Rabiner L.R. (1987) Signal Restoration by Spectral Mapping. *Proc. IEEE, Int. Conf. on Acoustics. Speech and Signal Processing*, **ICASSP-87** Texas: 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**: 833–837.

Lim J.S. (1978) Evaluations of Correlation Subtraction Method for Enhancing Speech Degraded by Additive White Noise. *IEEE Trans. Acoustics, Speech and Signal Processing*, **ASSP-26**(5): 471–472.

Linhard K., Klemm H. (1997) Noise Reduction with Spectral Subtraction and Median Filtering for Suppression of Musical Tones. Proc. ECSA-NATO Workshop on Robust Speech Recognition: 159–162.

Lockwood P., Boudy J. (1992) *Experiments with a Non-linear Spectral Subtractor (NSS) Hidden Markov Models and the Projection, for Robust Speech Recognition in Car, Speech Communications*. Elsevier: 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**: 265–268.

Milner B.P. (1995) Speech Recognition in Adverse Environments. PhD Thesis, University of East Anglia, UK.

Mcaulay R.J. and Malpass M.L. (1980) Speech Enhancement Using A Soft-Decision Noise Suppression Filter. *IEEE Trans*. **ASSP-28**(2): 137–145, April.

Nolazco-Flores J., Young S. (1993) Adapting a HMM-based Recogniser for Noisy Speech Enhanced by Spectral Subtraction, Proceedings Eurospeech: 829–832.

Porter J.E. and Boll S.F. (1984) Optimal Estimators for Spectral Restoration of Noisy Speech. *Proc. IEEE, Int. Conf. on Acoustics. Speech and Signal Processing*, **ICASSP-84**: 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**: 1073–1076.

Sorenson H.B. (1993) Robust Speaker Independent Speech Recognition Using Non-Linear Spectral Subtraction Based IMELDA. EuroSpeech-**93**: 235–238.

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.

Van Compernolle D. (1989) Noise Adaptation in a Hidden Markov Model Speech Recognition System. *Computer Speech and Language*, **3**: 151–167.

Vaseghi S.V. and Frayling-Corck R. (1992) Restoration of Archived Gramophone Records, *Journal of Audio Engineering Society*, **40**(10): 791–801, Oct.

Xie F. (1993) Speech Enhancement by Non-Linear Spectral Estimation a Unifying Approach. EuroSpeech-**93**: 617–620.

Zwicker E. and Fastel H. (1999) *Psychoacoustics, Facts and Models*, 2nd edn, Springer.

13

Impulsive Noise: Modelling, Detection and Removal

Impulsive noise consists of relatively short duration ‘on/off’ noise pulses, caused by a variety of interfering sources, channel effects or device defects, such as switching noise, interfering electromagnetic pulses, adverse communication channel environments, data packet loss, signal dropouts, physical degradation/scratch of the storage medium, 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 processes 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. A model-based impulsive noise removal method is described that detects each impulsive noise and then proceeds to replace the samples obliterated by an impulse. Some methods for introducing robustness to impulsive noise in parameter estimation are also considered.

13.1 Impulsive Noise

In this section, first, the mathematical concepts of an impulse function in continuous-time and discrete-time domains are introduced and then the various forms and shapes of real impulsive noise in communication systems are considered and the appropriate choice of the domain for the processing of impulsive noise are discussed.

13.1.1 Definition of a Theoretical Impulse Function

The formulation of the mathematical concept of an impulse in continuous-time domain (an analoge impulse) is illustrated in Figure 13.1. Consider the unit-area pulse $p(t)$ shown in Figure 13.1(a). As the pulse width Δ tends to zero, the pulse tends to an impulse with infinite amplitude, energy and power while the area under the pulse remains equal to unity.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

The impulse function shown in Figure 13.1(b) is defined as a pulse with an infinitesimal time width as

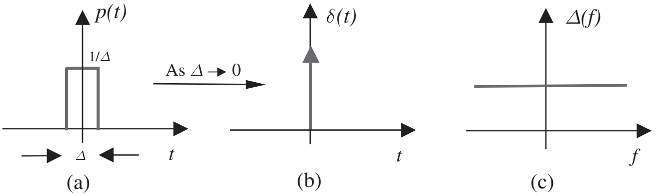


Figure 13.1 (a) A unit-area pulse, (b) The pulse becomes an impulse as its duration $\varDelta \to 0$, (c) The spectrum of the impulse function.

$$\delta(t) = \underset{\varDelta\to 0}{\text{limit}}\; p(t) = \begin{cases} 1/\varDelta, & |t| \le \varDelta/2 \\ 0, & |t| > \varDelta/2 \end{cases} \tag{13.1}$$

The integral of the impulse function is given by

$$\int_{-\infty}^{\infty} \delta(t)\, dt = \varDelta \times \frac{1}{\varDelta} = 1 \tag{13.2}$$

The Fourier transform of the impulse function is obtained as

$$\varDelta(f) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft} dt = e^0 = 1 \tag{13.3}$$

where f is the frequency variable. The impulse function is often used as a test function to obtain the impulse response of a system. This is because as shown by Equation (13.31) and in Figure 13.1(c), an impulse is a spectrally rich signal containing all frequencies in equal amounts.

A discrete-time or digital impulse $\delta(m)$, shown Figure 13.2(a), is defined as a signal with an 'on' duration of one sample, and is expressed as:

$$\delta(m) = \begin{cases} 1, & m = 0 \\ 0, & m \ne 0 \end{cases} \tag{13.4}$$

where the variable m designates the discrete-time index. Using the Fourier transform relation, the frequency spectrum of a digital impulse is given by

$$\varDelta(f) = \sum_{m=-\infty}^{\infty} \delta(m) e^{-j2\pi fm} = 1.0, \quad -\infty < f < \infty \tag{13.5}$$

13.1.2 The Shape of a Real Impulse in a Communication System

It should be noted that theoretical impulses of infinitesimal (infinitely small) duration, do not exist in real-life engineering systems that we normally experience not least because the realisation of a theoretical impulse would require immeasurable power as energy would be delivered in infinitesimally small time. Note further that even a discrete-time signal with a duration of only one sample has a finite duration of equal to the inverse of the sampling rate $1/F_s$.

Hence, in communication systems, real impulsive-type noise has a finite non-zero duration that, in sampled discrete-time form, is normally more than one sample long. For example, in the context of audio signals, short-duration, sharp pulses, of the order of 3 milliseconds (60 samples at a 20 kHz sampling rate) may be considered as impulsive-type noise. Figures 13.2(c) and 13.2(d) illustrate two examples of short-duration pulses and their respective spectra.

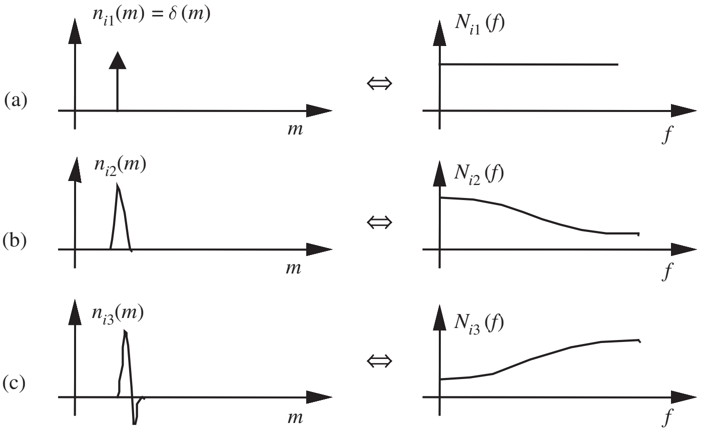


Figure 13.2 Time and frequency sketches of (a) an ideal impulse, and (b) and (c) two examples of short-duration pulses.

13.1.3 The Response of a Communication System to an Impulse

The signal observed as an impulsive-type noise is usually the response of a communication system and/or a recording sensor to an impulsive noise. The response of a system to an impulse is known as its impulse response. Figure 13.3(a) illustrates an impulsive or click-type noise modelled as the decaying oscillatory impulse response of a communication channel.

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 temporal-spectral shape of the received noise is affected by the channel, and can be considered as the channel impulse response.

In general, the response of a communication channel may be linear or non-linear, stationary or time varying. Furthermore, many communication systems, in response to a large-amplitude impulse, exhibit a nonlinear characteristic. Figure 13.3(b)–(d) illustrate some examples of impulsive noise, typical of those observed on an old gramophone recording. In this case, the communication channel is the electro-mechanical playback system composed of the combination of the stylus mechanism for reading the record grooves and the transducer-amplifier circuits for conversion of the mechanical movements of stylus to amplified electrical signals. It may be assumed that the response of the gramophone system is time-invariant. The figure also shows some variations of the channel characteristics with the amplitude of impulsive noise. These variations may be attributed to the non-linear characteristics of the playback mechanism.

13.1.4 The Choice of Time or Frequency Domain for Processing of Signals Degraded by Impulsive Noise

An important consideration in the development of a noise processing system is the choice of an appropriate domain for signal representation, which can be either time domain or frequency domain or multi-resolution time-frequency analysis as in wavelets. The choice should depend on the specific objective of the system.

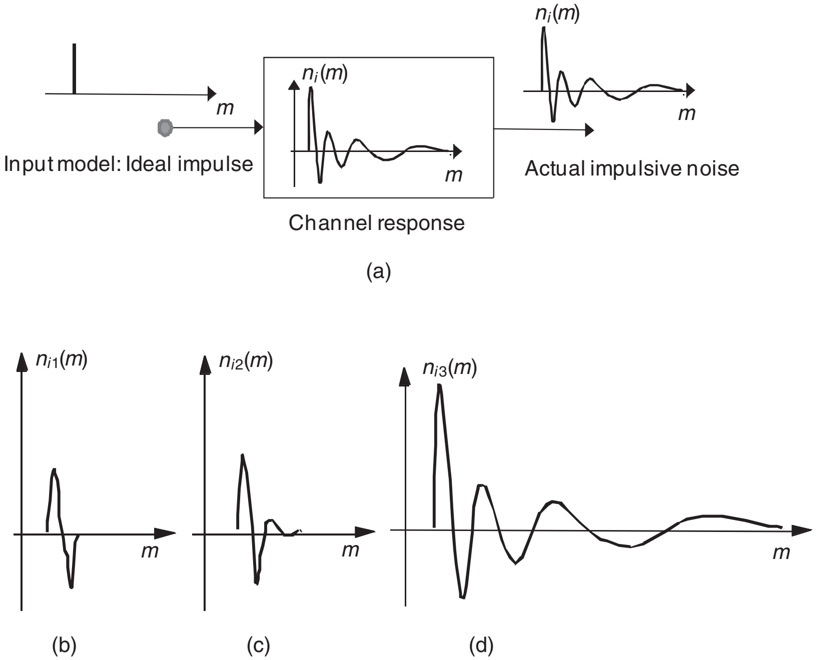


Figure 13.3 Illustrations of: (a) an actual impulsive-type (or click-type) noise modelled as the response of a filter to an ideal impulse, (b)–(d) the variations of the impulse response of a non-linear system with increasing amplitude of the impulse may have an impact on the duration of the pulse.

In signal restoration, the objective is to separate the noise from the signal, and the representation domain must be the one that emphasises 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.

13.2 Autocorrelation and Power Spectrum of Impulsive Noise

Impulsive noise is a non-stationary, binary-state sequence of impulses with random amplitudes and random times of occurrence. The non-stationary nature of impulsive noise can be seen by considering the power spectrum of an impulsive 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 functions of an impulsive noise are binary state, time-varying processes.

An impulsive noise sequence can be modelled as an amplitude-modulated binary-state sequence, and expressed as

$$n_i(m) = n(m)b(m) \tag{13.6}$$

where the state function $b(m)$ is a random binary sequence of ones (indicating impulse are on) and zeros (indicating impulses are off) 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:

$$r_{nn}(k,m) = \mathcal{E}[n_i(m)n_i(m+k)] = \sigma_n^2\delta(k)b(m) \tag{13.7}$$

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 (13.7) may be written as

$$r_{nn}(0,m) = \sigma_n^2 b(m) \tag{13.8}$$

Note that for a zero-mean noise process, $r_{nn}(0,m)$ is the time-varying binary-state noise power. The binary-state power spectrum of an impulsive noise sequence is obtained, by taking the Fourier transform of the autocorrelation function Equation (13.8), as

$$P_{N_I N_I}(f,m) = \sigma_n^2 b(m) \tag{13.9}$$

In Equations (13.8) and (13.9) the autocorrelation and power spectrum are expressed as binary state functions that depend on the 'on/off' state of impulsive noise at time m.

13.3 Probability Models of 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

$$n_i(m) = \sum_{k=0}^{P-1} h(k)n(m-k)b(m-k) \tag{13.10}$$

In Equation (13.10) $b(m)$ is a binary-valued random sequence model of the time of occurrence of impulsive 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. Figure 13.4 illustrates the impulsive noise model of Equation (13.10).

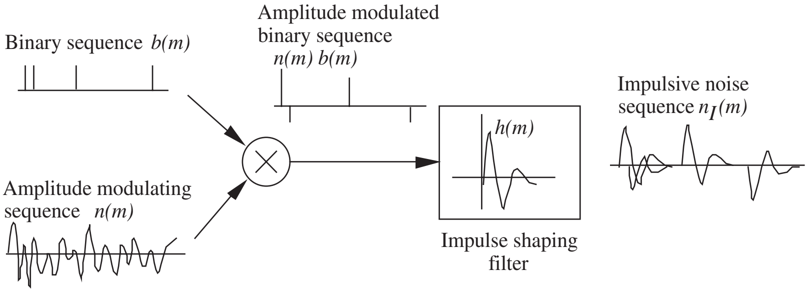

Binary sequence b(m)
Amplitude modulated binary sequence n(m) b(m)
Impulsive noise sequence $n_I(m)$
h(m)
Amplitude modulating sequence n(m)
Impulse shaping filter

Figure 13.4 Illustration of an impulsive noise model as the output of a filter excited by an amplitude-modulated binary sequence.

In the following part of this section four statistical processes for modelling an impulsive noise process as an amplitude-modulated binary sequence are considered. These are Bernoulli–Gaussian process, Poisson–Gaussian process, Binary state model and the hidden Markov model.

13.3.1 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 $1-\alpha$. In the context of modellin an impulse process a value of $b(m)=1$ signals the presence of an impulse whereas a value of $b(m)=0$ signals the absence of an impulse.

The probability mass function of a Bernoulli process is given by

$$P_B\left(b(m)\right)=\begin{cases}\alpha & \text{for} \quad b(m)=1\\ 1-\alpha & \text{for} \quad b(m)=0.\end{cases} \tag{13.11}$$

A Bernoulli process has a mean

$$\begin{aligned}\mu_b &= \mathcal{E}[(b(m))]=\alpha\times 1+(1-\alpha)\times 0\\ &=\alpha\end{aligned} \tag{13.12}$$

and a variance

$$\begin{aligned}\sigma_b^2 &= \mathcal{E}\left[(b(m)-\mu_b)^2\right]=\mathcal{E}\left[b(m)^2\right]-2\mathcal{E}[b(m)]\mu_b+{\mu_b}^2\\ &=\alpha(1-\alpha)\end{aligned} \tag{13.13}$$

A zero-mean Gaussian pdf model of the random amplitudes of impulsive noise is given by

$$f_N\big(n(m)\big)=\frac{1}{\sqrt{2\pi}\,\sigma_n}\exp\left[-\frac{n^2(m)}{2\sigma_n^2}\right] \tag{13.14}$$

where σ_n^2 is the variance of the noise amplitude.

In a Bernoulli–Gaussian model the probability density function of an impulsive noise $\mathrm{n_i(m)}$ is given by a mixture of two probabilities as

$$f_N^{BG}\big(n_i(m)\big)=(1-\alpha)\underbrace{\delta\big(n_i(m)\big)}_{\substack{\text{Dirac pdf models}\\ \text{absence of noise}}}+\alpha\underbrace{f_N\big(n_i(m)\big)}_{\substack{\text{Gaussian pdf models}\\ \text{amplitude of noise}}} \tag{13.15}$$

where $\delta\big(n_i(m)\big)$ is the Kronecker delta pdf (or dirac pdf) that models the absence of noise i.e. $\delta\big(n_i(m)\big)$ is non-zero when its argument $n_i(m)$ zero. Note that the function $f_N^{BG}\big(n_i(m)\big)$ is a mixture of a discrete probability mass function $\delta\big(n_i(m)\big)$ and a continuous probability density function $f_N\big(n_i(m)\big)$.

An alternative model for impulsive noise is a binary-state Gaussian process (Section 3.7.4), with a low-variance state modelling the absence of impulses and a relatively high-variance state modelling the amplitude of impulsive noise.

13.3.2 Poisson–Gaussian Model of Impulsive Noise

In a Poisson–Gaussian model the probability of occurrence of a number of impulsive noise events, in a time interval of T seconds, 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 3, is a random event-counting process. In a Poisson model, the probability of occurrence of k impulsive noise in a time interval of T is given by

$$P(k,T) = \frac{(\lambda T)^k}{k!} e^{-\lambda T} \tag{13.16}$$

Where λT is the mean value of the process that is the average number of impulses that occur in a time interval of T and the parameter λ is a rate function with the following properties:

$$\textit{Prob}\text{ (one impulse in a small time interval } \Delta t) = \lambda \Delta t \tag{13.17}$$

$$\textit{Prob}\text{ (zero impulse in a small time interval } \Delta t) = 1 - \lambda \Delta t \tag{13.18}$$

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 of Δt is given by

$$f_{N_I}^{PG}\big(n_i(m)\big) = (1 - \lambda \Delta t)\,\delta\big(n_i(m)\big) + \lambda \Delta t\, f_N\big(n_i(m)\big) \tag{13.19}$$

where $f_N(n_i\ (m))$ is the Gaussian pdf of Equation (13.14).

Also note that from Equation (13.16) we can deduce (as shown in Section 3.7.5) that mean and variance of the number of impulses in a time interval of T are given by

$$\text{Expected (mean) number of impulses in } T \text{ seconds} = \lambda T \tag{13.20}$$

$$\text{Variance of number of occurances of impulses in } T \text{ seconds} = \lambda T \tag{13.21}$$

13.3.3 A Binary-State Model of Impulsive Noise

An impulsive noise process may be modelled by a binary-state model as shown in Figure 13.5. In this binary model, the state S_0 corresponds to the ‘impulse off’ condition when impulsive noise is absent; in this state, the model emits zero-valued samples or the background noise. The state S_1 corresponds to the ‘impulse 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}.

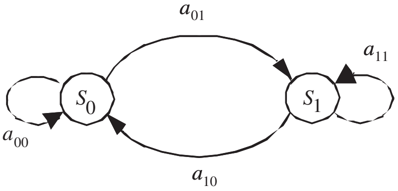


Figure 13.5 A binary-state model of an impulsive noise generator.

In its simplest form, as shown in Figure 13.5, the model is memory-less; meaning that 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

$$P\big(s(t+1) = S_0\,|s(t) = S_0\big) = P\big(s(t+1) = S_0\,|s(t) = S_1\big) = 1 - \alpha \tag{13.22}$$

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

$$P\big(s(t+1) = S_1\,|s(t) = S_0\big) = P\big(s(t+1) = S_1\,|s(t) = S_1\big) = \alpha \tag{13.23}$$

In one of its simplest forms, the impulse emitting state S_1 emits samples from a zero-mean Gaussian random process. However, 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 code book of M prototype impulsive noises, and their associated probabilities $[(n_{i1}, p_{i1}), (n_{i2}\, p_{i2}), \ldots, (n_{iM}, p_{iM})]$, 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 13.6 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.

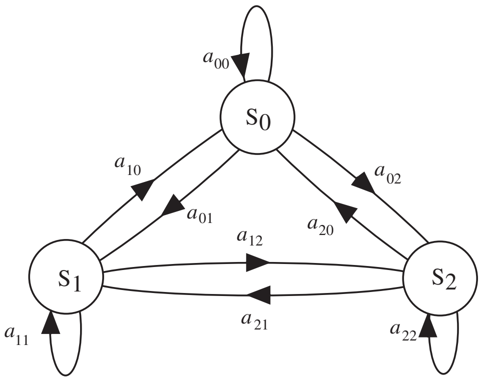


Figure 13.6 A 3-state model of impulsive noise and the decaying oscillations that often follow the impulses.

13.3.4 Hidden Markov Model of Impulsive and Burst Noise

Hidden Markov models (HMMs) described in Chapter 5 can be used to model impulsive noise processes. HMMs are defined by two sets of parameters, the Markovian state transition probabilities $\{a_{ij}\}$ and the state observation probabilities $\{b_{ik}\}$. The state transition probability models the number and pattern of occurrences of the impulses whereas the state observation probability models the amplitude of the impulses. A popular model for state observation probability is a Gaussian mixture model.

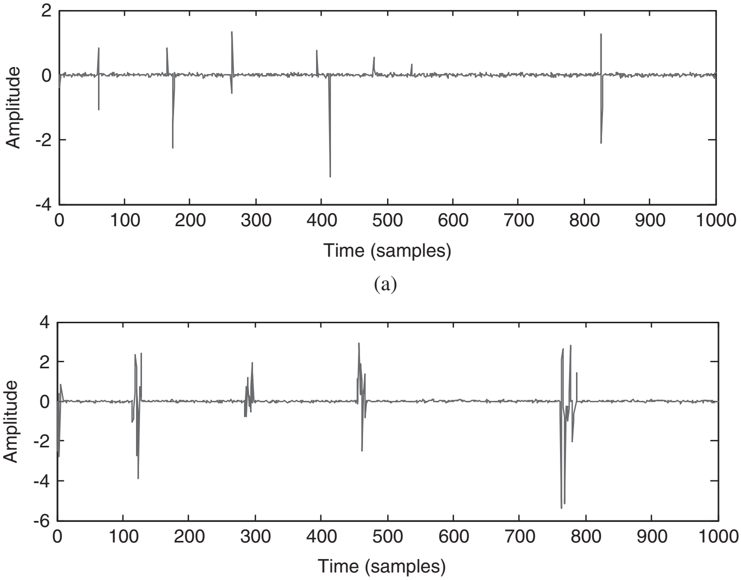
The state transition probabilities can affect a variety of different statistical patterns of the intervals of occurrences and the durations of impulsive noise. For example, consider a binary-state HMM structure, as shown in Figure 13.5, and assume that the state S_0 models the absence of impulses and emits zeros or a low level background noise and the state S_1 models the presence of impulses. The self-loop transition probability of S_0, a_{00}, can be used to control the duration of the impulse-absent intervals (i.e. the interval between the emissions of impulses), whereas the self-loop transition probability of S_1, a_{11}, can be used to control the individual or burst nature of impulses emitted in state S_1. The probabilities of the states S_0 and S_1, a measure of the average number of time spent in the states, are given by

$$P_0 = P_0 a_{00} + (1 - P_0) a_{10} = \frac{a_{10}}{1 + a_{00} + a_{10}} \tag{13.24}$$

$$P_1 = P_1 a_{11} + (1 - P_1) a_{01} = \frac{a_{01}}{1 + a_{11} + a_{01}} \tag{13.25}$$

The random interval of the onset of pulses is controlled by the probability of exit from state S_0 and is proportional to $1/a_{01}$.

Figure 13.7 illustrates two examples of impulsive noise generated by a two-state HMM. As shown a value of $a_{11} = 0.5$ produces isolated impulses whereas a value of a_{11} close to 1 produces bursts of impulses. Through the variation of the choice of transition probabilities the percentage of impulsive noise and their duration and burst nature can be varied.



(a)

(b)

Figure 13.7 Impulsive noise generated by a two-state HMM: (a) State S_0 emits background noise with a self-loop probability of $a_{00} = 0.98$ and S_1 emits samples from a Gaussian pdf with a self-loop probability of $a_{11} = 0.5$. (b) same as (a) only $a_{11} = 0.95$, hence state S_1 emits bursts of noise instead of impulse-type noise.

13.4 Impulsive Noise Contamination, Signal to Impulsive Noise Ratio

Several different measures can be used to quantify the degradation caused by impulsive noise. These include the percentage of signal samples contaminated by impulses, the mean and variance of the interval of impulses, the average signal to noise ratio and the local signal to noise ratio. When the impulsive noise happens in burst form then additional parameters required are the means and variances of the number of impulses in a burst, the duration of bursts and the burst interval.

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

$$SINR(m) = \frac{P_{\text{signal}}(m)}{P_{\text{impulse}}\, b(m)} \tag{13.26}$$

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

$$SINR = \frac{P_{\text{signal}}}{\alpha \; P_{\text{impulse}}} \tag{13.27}$$

Note that from Equation (13.27), for a given signal power, there are many pair of values of α and P_{Impulse} that can yield the same average SINR.

13.5 Median Filters for Removal of Impulsive Noise

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 are larger than $x_{\text{med}}(m)$ and half are smaller than $x_{\text{med}}(m)$. Hence the median of a set of samples is obtained by sorting the samples in the 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 13.8.

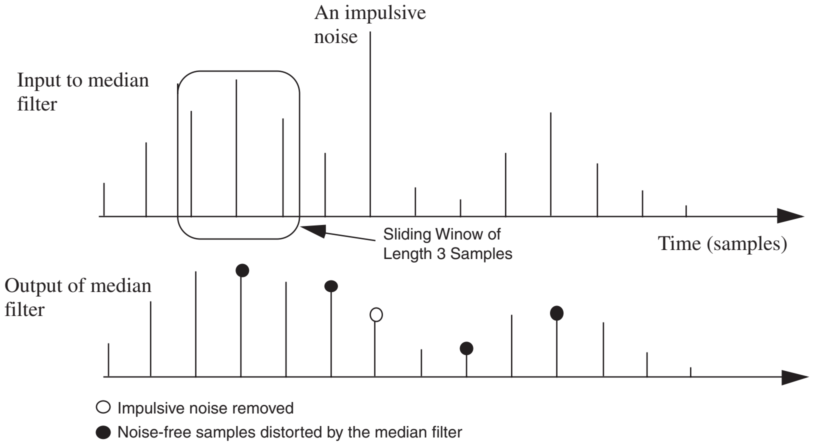

An impulsive noise
Input to median filter
Sliding Winow of Length 3 Samples
Time (samples)
Output of median filter
○ Impulsive noise removed
● Noise-free samples distorted by the median filter

Figure 13.8 Input and output of a median filter. Note that in addition to suppressing the impulsive outlier, the filter also distorts some genuine signal components when it swaps the sample value with the median.

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

$$\begin{aligned}\hat{x}(m) &= y_{\text{med}}(m)\\ &= \text{median}\,[y(m-K), \ldots,\; y(m), \ldots,\; y(m+K)]\end{aligned} \tag{13.28}$$

The median of a set of numbers is a non-linear statistics of the set, with the useful property that it is insensitive to the presence of a sample with an unusually large value, a so-called outlier, 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 effectively with 'real' impulsive noise, which are 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:

$$\hat{x}(m) = \begin{cases} y(m) & \text{if } |y(m) - y_{\text{med}}(m)| < k\theta(m) \\ y_{\text{med}}(m) & \text{otherwise} \end{cases} \tag{13.29}$$

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.

13.6 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 usually a relatively large fraction, $1 - \alpha$, of the signal samples remain 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 are not contaminated by impulsive noise.

The impulsive noise removal system shown in Figure 13.9 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. Both the detector and the interpolator share the linear prediction analysis system.

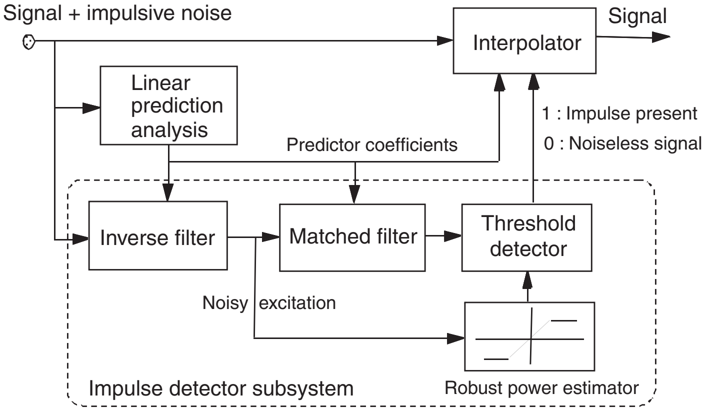


Figure 13.9 Configuration of an impulsive noise removal system incorporating a detector and interpolator subsystems.

The detector is composed of an inverse linear prediction analysis system, a matched filter and a threshold detector. The function of the inverse linear prediction filter is to whiten the signal. The visibility and detection of discontinuities such as impulsive noise is enhanced by application of the inverse linear prediction filter.

The output of the impulsive noise detector is a binary switch that 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.

13.6.1 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 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 13.10 shows a model for a noisy signal. The noise-free signal $x(m)$ is described by a linear prediction model as

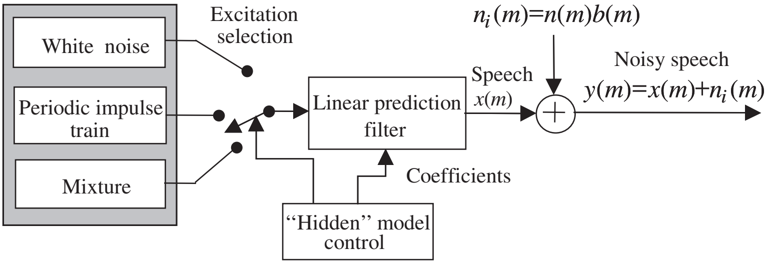


Figure 13.10 Noisy speech model. The signal is modelled by a linear predictor. Impulsive noise is modelled as an amplitude-modulated binary-state process.

$$x(m) = \sum_{k=1}^{P} a_k\, x(m-k) + e(m) \tag{13.30}$$

where $\boldsymbol{a} = [a_1, a_2, \ldots, a_P]^T$ 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 quasi-periodic train of pulses as illustrated in Figure 13.10. 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 delectability. Speech or music is composed of random excitations spectrally shaped and amplified by the resonances of 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.

13.6.2 Analysis of Improvement in Noise Detectability

In the following, the improvement in noise detectability that results from inverse filtering is analysed. Using Equation (13.30), we can rewrite a noisy signal model as

$$\begin{aligned} y(m) &= x(m) + n_i(m) \\ &= \sum_{k=1}^{P} a_k\, x(m-k) + e(m) + n_i(m) \end{aligned} \tag{13.31}$$

where $y(m)$, $x(m)$ and $n_i(m)$ are the noisy signal, the signal and the noise respectively. Using an estimate $\hat{\boldsymbol{a}}$ of the predictor coefficient vector $\boldsymbol{a}$, the noisy signal $y(m)$ can be inverse-filtered and transformed to the noisy excitation signal $v(m)$ as

$$\begin{aligned} v(m) &= y(m) - \sum_{k=1}^{P} \hat{a}_k\, y(m-k) \\ &= x(m) + n_i(m) - \sum_{k=1}^{P} (a_k - \tilde{a}_k)[x(m-k) + n_i(m-k)] \end{aligned} \tag{13.32}$$

where $\tilde{a}_k$ is the error in the estimate of the predictor coefficient. Using Equation (13.30) Equation (13.32) can be rewritten in the following form:

$$v(m) = e(m) + n_i(m) + \sum_{k=1}^{P} \tilde{a}_k x(m-k) - \sum_{k=1}^{P} \hat{a}_k n_i(m-k) \tag{13.33}$$

From Equation (13.33) there are essentially three terms that contribute to the noise in the excitation sequence:

(1) the impulsive disturbance $ni(m)$ which is usually the dominant noise 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

$$\frac{\text{Impulsive noise power}}{\text{Signal power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[x^2(m)]} \tag{13.34}$$

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

$$\frac{\text{Impulsive noise power}}{\text{Excitation power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[e^2(m)]} \tag{13.35}$$

The overall gain in impulsive noise to signal ratio (INSR) is obtained, by dividing Equations (13.34) and (13.35), as

$$INSR \text{ gain} = \frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]} \tag{13.36}$$

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} of that of the speech itself; therefore substantial improvement in impulsive noise detectability can be expected through inverse filtering of the noisy speech signals.

Figure 13.11 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 $[1, -a_1, \ldots, -a_P]$, as indicated by the term $\sum \hat{a}_k n_i(m-k)$ in Equation (13.33). 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.

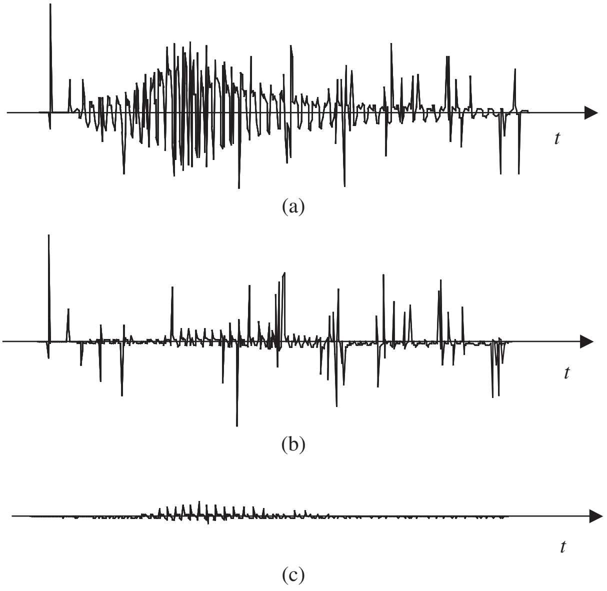


Figure 13.11 Illustration of the effects of inverse filtering on detectability of Impulsive noise: (a) Impulsive noise contaminated speech with 5 % impulse contamination at an average SINR of 10dB, (b) Speech excitation of impulse contaminated speech, and (c) Speech excitation of impulse-free speech.

13.6.3 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 of a two-sided linear prediction model. The two-sided linear prediction of a sample $\boldsymbol{x}(m)$ is based on the P past samples and the P future samples, and is defined by the equation

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + \sum_{k=1}^{P} a_{k+P}\, x(m+k) + e(m) \tag{13.37}$$

where a_k are the two-sided predictor coefficients and $e(m)$ is the excitation signal. All the analysis used for the case of 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 (13.33) the correlations of each sample with the future, as well as the past, samples are modelled. Although Equation (13.33) is a non-causal 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.

13.6.4 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 are introduced in Chapter 11. 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.

13.7 Robust Parameter Estimation

In the block diagram of Figure 13.9, 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 robust estimation of a parameter, such as the signal power, in the presence of impulsive noise.

A robust estimator is one that is not over-sensitive 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 13.12 illustrates the variations of several cost-of-error functions with a parameter θ. Figure 13.12(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 13.12(a) that an unbounded sample has an unbounded influence on the estimation results.

A method for introducing robustness is to use a non-linear 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 13.12(b). One disadvantage of this function is that it is not continuous

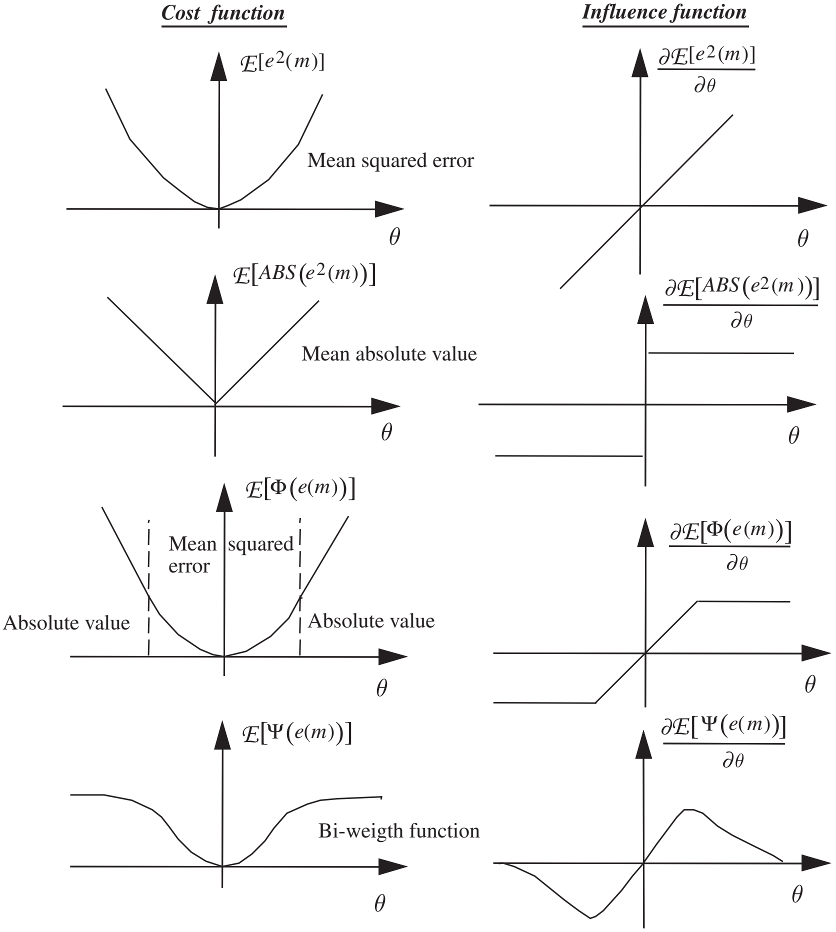


Figure 13.12 Illustration of a number of cost-of-error functions and the corresponding influence functions.

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

$$\psi[e(m)] = \begin{cases} e^2(m) & \text{if } |e(m)| \le k \\ k\,|e(m)| & \text{otherwise} \end{cases} \tag{13.38}$$

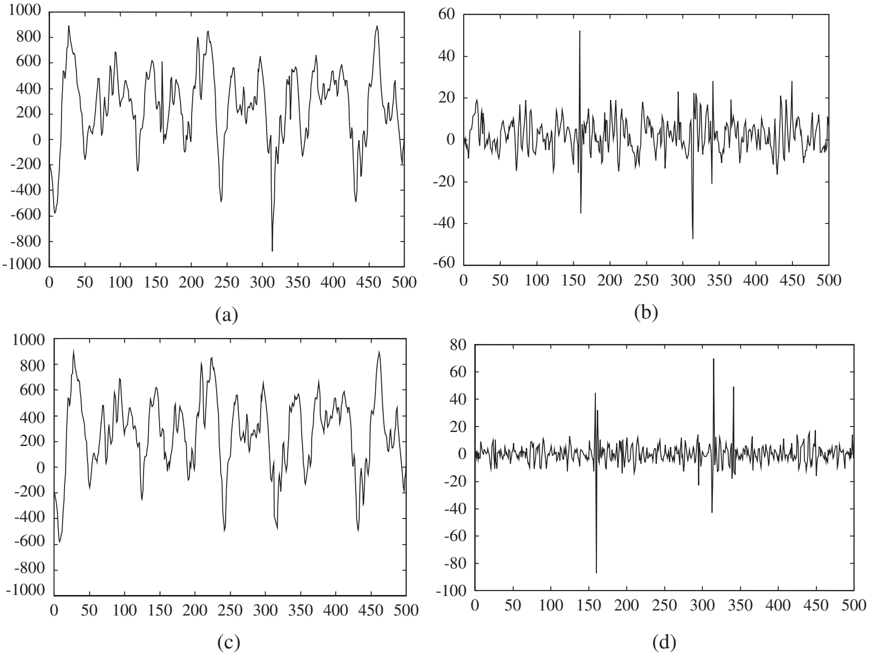
Huber's function, shown in Figure 13.12(c), is a hybrid of the least mean square and the absolute value of error functions. Tukey's bi-weight function, which is a redescending robust objective function, is defined as

$$\psi[e(m)] = \begin{cases} \{1 - [1 - e^2(m)]^3\}/6, & \text{if } |e(m)| \le 1 \\ 1/6 & \text{otherwise} \end{cases} \tag{13.39}$$

As shown in Figure 13.12(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.

13.8 Restoration of Archived Gramophone Records

This Section describes the application of the impulsive noise removal system of Figure 13.9 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 13.13(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 Figures 13.13(b) and (c) respectively. Close examination of these figures show that some of the ambiguities between the noise pulses and the genuine signal excitation pulses are resolved after matched filtering.



(a)

(b)

(c)

(d)

Figure 13.13 (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 11. Figure 13.13(d) shows the signal with the impulses removed. The impulsive noise removal system of Figure 13.9 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.

13.9 Summary

The classic linear time-invariant theory on which many signal processing methods are based is not suitable for dealing with the non-stationary 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 model, the Poisson–Gaussian and the hidden Markov model (HMM). 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 operation enhance the detectibility of impulsive noise. Based on this observation, in Section 13.6, 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 in this chapter 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. *Journal of the Royal Statistical Society*, Ser. **39**: 1–38.

Godsil S. (1993) *Restoration of Degraded Audio Signals*. Cambridge University Press.

Gallagher N.C. and Wise G.L. (1981) A Theoretical Analysis of the Properties of Median Filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, **ASSP-29**: 1136–1141.

Jaynat N.S. (1976) Average and Median Based Smoothing for Improving Digital Speech Quality in the Presence of Transmission Errors. *IEEE Trans. Commun*: 1043–1045, Sept.

Kelma V.C. and Laub A.J. (1980) The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Trans. Automatic 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. Acoustics, Speech and Signal Processing*, **ASSP, 32**(3): 600–609, June.

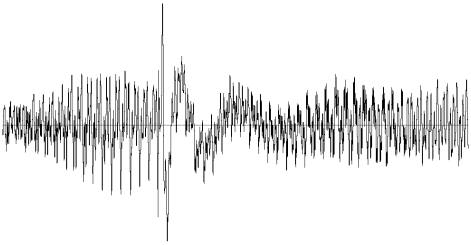
Nieminen, Heinonen P. and Neuvo Y. (1987) Suppression and Detection of Impulsive Type Interference using Adaptive Median Hybrid Filters. *IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing*, **ICASSP-87**: 117–120.

Tukey J.W. (1971) *Exploratory Data Analysis*. Addison Wesley, Reading, MA.

Rabiner L.R., Sambur M.R. and Schmidt C.E. (1984) Applications of a Nonlinear Smoothing Algorithm to Speech Processing. *IEEE Trans*. **ASSP-32**, 3, June.

Vaseghi S.V. and Rayner P.J.W. (1990) Detection and Suppression of Impulsive Noise in Speech Communication Systems. *IEE Proc-I Communications Speech and Vision*: 38–46, February.

Vaseghi S.V. and Milner B.P. (1995) Speech Recognition in Impulsive Noise, Inst. of Acoustics, Speech and Signal Processing. *IEEE Proc. Int. Conf. Acoustics, Speech and Signal Processing*, **ICASSP-95**: 437–440.



14

Transient Noise Pulses

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 13 on impulsive noise.

14.1 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 14.1. 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 telecommunication 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 characterise the transient noise pulses with a similar degree of consistency to that of characterising 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 14.1 and 14.2. Scratch noise pulses are acoustic manifestations of the response of the stylus and the associated electro-mechanical 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

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

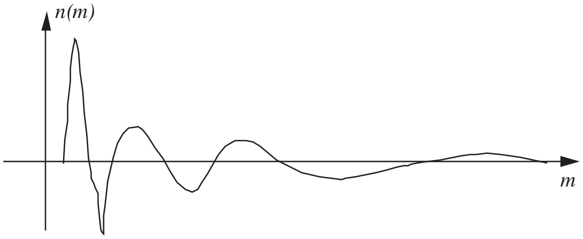


Figure 14.1 The profile of a transient noise pulse from a scratched gramophone record.

exhibit a similar characteristics. As shown in Figure 14.1, 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 14.1 that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the nonlinear modes of response of the electro-mechanical 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 14.2(a) and (b) show the time-domain waveform and the

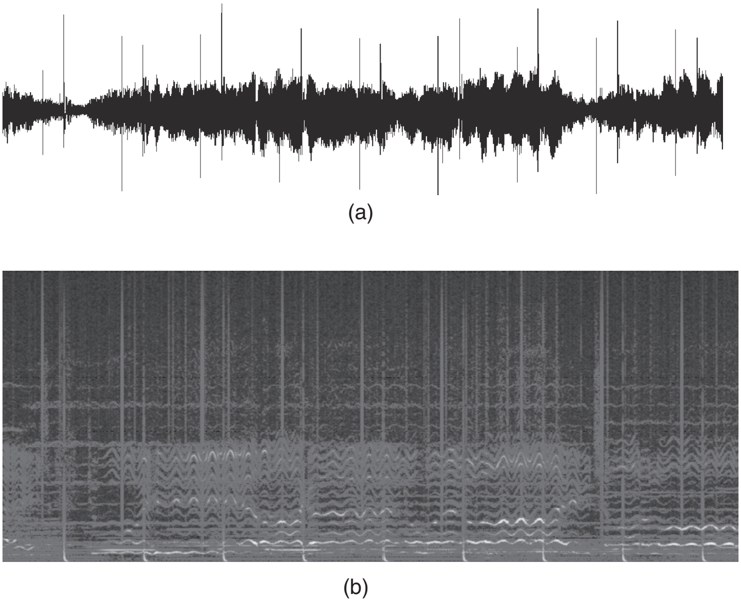


Figure 14.2 An example of (a) the time-domain waveform and (b) the spectrogram of transient noise scratch pulses in a damaged gramophone record.

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.

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.

14.2 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

$$n(m) = \sum_k h_k A\delta(m-k) = Ah_m \tag{14.1}$$

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

$$n(m) = \sum_k h_k \sum_j A_j \delta\left((m-T_j)-k\right) = \sum_j A_j h_{m-T_j} \tag{14.2}$$

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.

14.2.1 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 code book 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 code book 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 14.3 shows a number of typical noise pulses. The design of a code book of signal templates is described in Chapter 4.

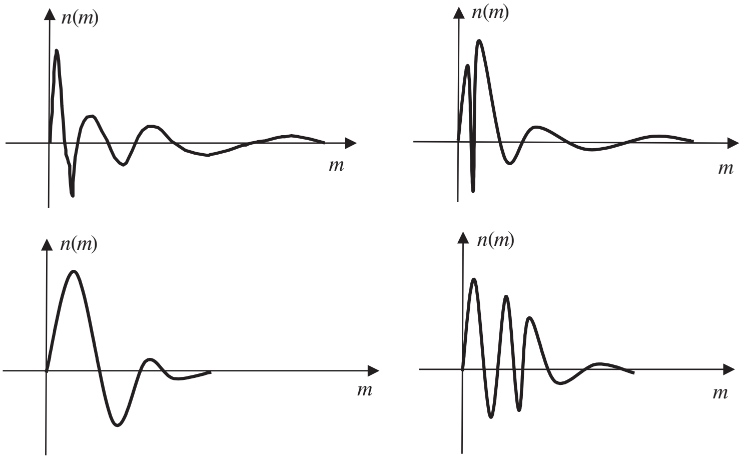


Figure 14.3 Illustration of a number of prototype transient pulses.

14.2.2 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 (AR) model of transient noise pulses. The AR model for a single noise pulse $n(m)$ can be described as

$$n(m) = \sum_{k=1}^{P} c_k n(m-k) + A\delta(m) \tag{14.3}$$

where c_k are the AR model coefficients, P is the AR model order 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:

$$n(m) = \sum_{k=1}^{P} c_k n(m-k) + \sum_{j}^{M} A_j\delta(m-T_j) \tag{14.4}$$

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 $\sigma_1^2 >> \sigma_0^2$. 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

$$e_n(m) = \left[\sigma_1 b(m) + \sigma_0 \bar{b}(m)\right] u(m) \tag{14.5}$$

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

$$\sigma_{e_n}^2(m) = \sigma_1^2 b(m) + \sigma_0^2 \bar{b}(m) \tag{14.6}$$

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 $\boldsymbol{n}$ is given by

$$f_N(\boldsymbol{n}) = \frac{1}{(2\pi)^{N/2}\left|\boldsymbol{\Lambda}_{e_n e_n}\right|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{e_n e_n}^{-1}\boldsymbol{C}\,\boldsymbol{n}\right) \tag{14.7}$$

where $\boldsymbol{C}$ is a matrix of coefficients of the AR model of the noise (as described in Section 8.4), and $\boldsymbol{\Lambda}_{e_n e_n}$ is the diagonal covariance matrix of the input to the noise model. The diagonal elements of $\boldsymbol{\Lambda}_{e_n e_n}$ are given by Equation (14.6).

14.2.3 Hidden Markov Model of a Noise Pulse Process

A hidden Markov model (HMM), described in Chapter 5, is a finite state statistical model for non-stationary 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 14.4 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 14.4, 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 non-linear pulse or the decaying oscillatory part.

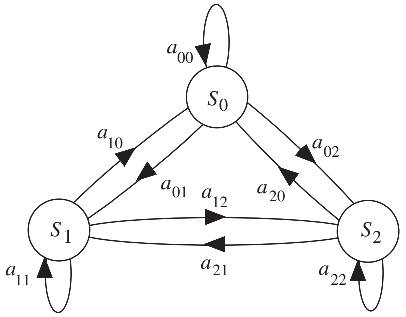


Figure 14.4 A three-state model of a transient noise pulse process.

A hidden Markov model of noise can be employed for both the detection and the removal of transient noise pulses. As described in Section 14.3.3, 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.

14.3 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

$$y(m) = b(m)n(m) + x(m) \tag{14.8}$$

where $b(m)$ is a binary 'indicator' process that signals the presence or absence of a noise pulse. Using the model of Equation (14.8), 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 14.2.

14.3.1 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

$$H(f) = K\frac{N^*(f)}{P_{XX}(f)} \tag{14.9}$$

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

$$H(f) = \frac{K}{\sigma_x^2}N^*(f) \tag{14.10}$$

The impulse response of the matched filter corresponding to Equation (14.10) is given by

$$h(m) = Cn(-m) \tag{14.11}$$

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

$$z(m) = Cn(-m) * n(m) \tag{14.12}$$

where the asterisk $*$ denotes convolution. In the frequency domain Equation (14.12) becomes

$$Z(f) = N(f)H(f) = C|N(f)|^2 \tag{14.13}$$

The matched filter output $z(m)$ is passed through a non-linearity and a decision is made on the presence or the absence of a noise pulse as

$$\hat{b}(m) = \begin{cases} 1 & \text{if } |z(m)| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{14.14}$$

In Equation (14.14), when the matched filter output exceeds a threshold, the detector flags the presence of the signal at the input. Figure 14.5 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 provide 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.

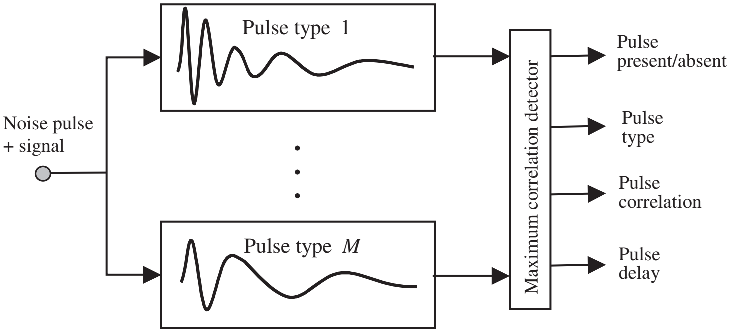


Figure 14.5 A bank of matched filters for detection of transient noise pulses.

This information can be used for subtraction of the noise pulse from the noisy signal as described in Section 14.4.1.

14.3.2 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 detectibility 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.

14.3.3 Noise Detection Based on HMM

In the three-state hidden Markov model of a transient noise pulse process, described in Section 14.2.3, 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 $\boldsymbol{y} = [y(0), y(1), \ldots, y(N-1)]$, the maximum likelihood state sequence $\boldsymbol{s} = [s(0), s(1), \ldots, s(N-1)]$, of the HMM $\mathcal{M}$ is obtained as

$$\boldsymbol{s}_{ML} = \arg\max_{s} f_{Y|S}(\boldsymbol{y}|\boldsymbol{s}, \mathcal{M}) \tag{14.15}$$

where, for a hidden Markov model, the likelihood of an observation sequence $f_{Y|S}(\boldsymbol{y}|\boldsymbol{s}, \lambda)$ can be expressed as

$$\begin{aligned} & f_{Y|S}\,(y(0), y(1), \ldots, y(N-1)\,|s(0), s(1), \ldots, s(N-1)\,) \\ & = \pi_{s(0)} f_{s(0)}\big(y(0)\big).a_{s(0),s(1)} f_{s(1)}\big(y(1)\big) \cdots a_{s(N-2),s(N-1)} f_{s(N-1)}\big(y(N-1)\big) \end{aligned} \tag{14.16}$$

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.

14.4 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 (AR) 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 the states of the noise pulse

14.4.1 Adaptive Subtraction of Noise Pulses

The transient noise removal system shown in Figure 14.6 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 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

$$y(m) = x(m) + b(m)n(m) \tag{14.17}$$

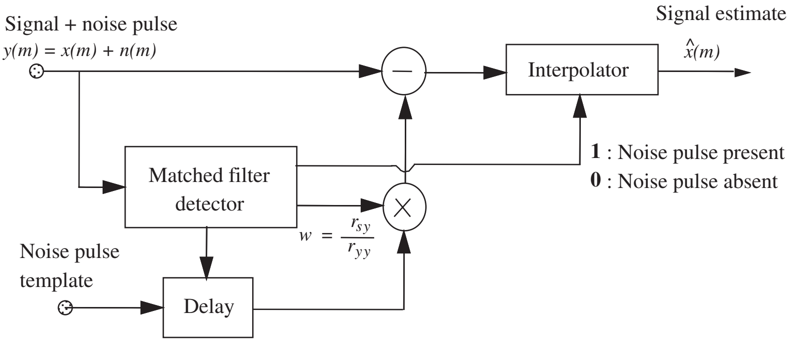


Figure 14.6 Transient noise pulse removal system.

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

$$n(m) \approx w\bar{n}(m - D) \tag{14.18}$$

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 (14.17) and (14.18) the noisy signal can be modelled:

$$y(m) \approx x(m) + w\bar{n}(m - D) \tag{14.19}$$

From Equation (14.19) an estimate of the signal $x(m)$ can be obtained by subtracting an estimate of the noise pulse from that of the noisy signal:

$$\hat{x}(m) = y(m) - w\bar{n}(m - D) \tag{14.20}$$

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

$$D = \arg\max_{k} \left[CCF\left(y(m), \bar{n}(m-k)\right)\right] \tag{14.21}$$

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

$$\begin{aligned}\sum_{m=0}^{N-1} y(m)\bar{n}(m-D) &= \sum_{m=0}^{N-1} [x(m) + w\bar{n}(m-D)]\bar{n}(m-D)\\ &= \sum_{m=0}^{N-1} x(m)\bar{n}(m-D) + w\sum_{m=0}^{N-1} \bar{n}(m-D)\bar{n}(m-D)\end{aligned} \tag{14.22}$$

where N is the pulse template length. Since the signal $x(m)$ and the noise $n(m)$ are uncorrelated, the term $\boldsymbol{\Sigma}x(m)\bar{n}(m-D)$ on the right hand side of Equation (14.22) is small, and we have

$$w \approx \frac{\sum_{m} y(m)\bar{n}(m-D)}{\sum_{m} \bar{n}^2(m-D)} \tag{14.23}$$

Note 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 11. When there is no noise pulse, the coefficient w is zero, the interpolator is bypassed and the input signal is passed through unmodified. Figure 14.7(b) shows the result of processing the noisy signal of Figure 14.7(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.

14.4.2 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

$$x(m) = \sum_{k=1}^{P_1} a_k x(m-k) + e(m) \tag{14.24}$$

Assuming that $e(m)$ is a zero-mean uncorrelated Gaussian process with variance σ_e^2, the pdf of a vector $\boldsymbol{x}$ of N successive signal samples of an autoregressive process with parameter vector $\boldsymbol{a}$ is given by

$$f_X(\boldsymbol{x}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \tag{14.25}$$

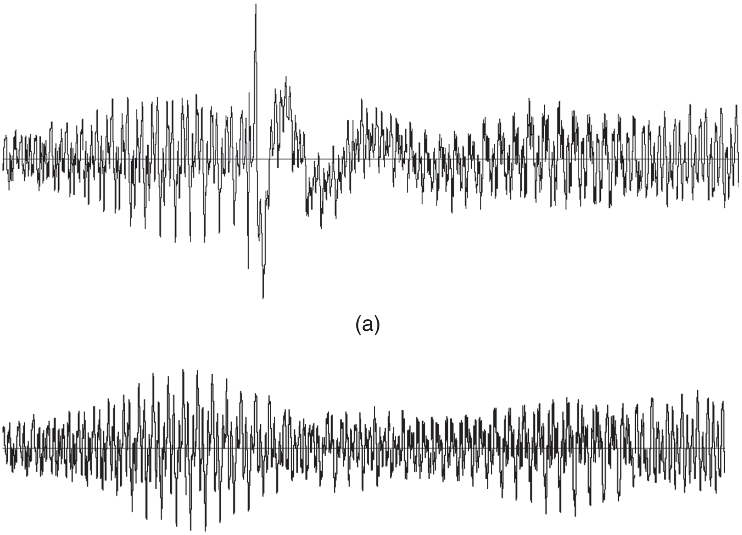


Figure 14.7 (a) A signal from an old gramophone record with a scratch noise pulse. (b) The restored signal.

where the elements of the matrix $\boldsymbol{A}$ are composed of the coefficients a_k of the linear predictor model as described in Section 8.4. In Equation (14.25), 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

$$n(m) = \sum_{k=1}^{P_2} c_k n(m-k) + A\delta(m) \tag{14.26}$$

where c_k are the model coefficients, P_2 is the model order, and the excitation is a assumed to be an impulse of amplitude A. A number of closely spaced and overlapping noise pulses can be modelled as

$$n(m) = \sum_{k=1}^{P_2} a_k n(m-k) + \sum_{j}^{M} A_j \delta(m-T_j) \tag{14.27}$$

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 $\sigma_1^2 >> \sigma_0^2$. 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

$$e_n(m) = \left[b(m)\sigma_1 + \bar{b}(m)\sigma_0\right] u(m) \tag{14.28}$$

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

$$\sigma_{e_n}^2(m) = b(m)\sigma_1^2 + \bar{b}(m)\sigma_0^2 \tag{14.29}$$

Assuming that the excitation state pattern $\boldsymbol{b} = [b(m)]$ is given, the pdf of an N sample noise pulse $\boldsymbol{x}$ is

$$f_N(\boldsymbol{n}|\boldsymbol{b}) = \frac{1}{(2\pi)^{N/2}\left|\boldsymbol{\Lambda}_{\boldsymbol{e_n e_n}}\right|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}_{\boldsymbol{e_n e_n}}\boldsymbol{Cn}\right) \tag{14.30}$$

where the elements of the matrix $\boldsymbol{C}$ are composed of the coefficients c_k of the linear predictor model as described in Section 8.4. The posterior pdf of the signal $\boldsymbol{x}$ given the noisy observation $\boldsymbol{y}$, $f_{X|Y}(\boldsymbol{x}|\boldsymbol{y})$,can be expressed, using Bayes' rule, as

$$\begin{aligned} f_{X|Y}(\boldsymbol{x}|\boldsymbol{y}) &= \frac{1}{f_Y(\boldsymbol{y})} f_{Y|X}(\boldsymbol{y}|\boldsymbol{x}) f_X(\boldsymbol{x}) \\ &= \frac{1}{f_Y(\boldsymbol{y})} f_N(\boldsymbol{y}-\boldsymbol{x}) f_X(\boldsymbol{x}) \end{aligned} \tag{14.31}$$

For a given observation $f_Y(\boldsymbol{y})$ is a constant. Substitution of Equations (14.30) and (14.25) in Equation (14.31) yields

$$\begin{aligned} f_{X|Y}(\boldsymbol{x}\,|\boldsymbol{y}\,) = &\frac{1}{f_Y(\boldsymbol{y})} \frac{1}{(2\pi\sigma_e)^N \left|\boldsymbol{\Lambda}_{e_n e_n}\right|^{1/2}} \\ &\times \exp\left(-\frac{1}{2}\,(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\,\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}_{e_n e_n}\boldsymbol{C}\,(\boldsymbol{y}-\boldsymbol{x}) - \frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}\right) \end{aligned} \tag{14.32}$$

The MAP solution obtained by maximisation of the log posterior function with respect to the undistorted signal $\boldsymbol{x}$ is given by

$$\hat{\boldsymbol{x}}^{MAP} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\big/\sigma_e^2 + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}_{e_n e_n}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}_{e_n e_n}\boldsymbol{Cy} \tag{14.33}$$

where $\hat{\boldsymbol{x}}^{MAP}$ is the MAP interpolation.

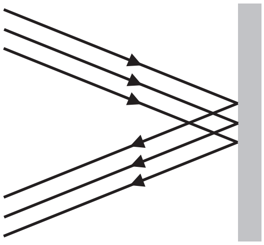
14.5 Summary

In this chapter, we considered the modelling, detection and removal of transient noise pulses. Transient noise pulses are non-stationary 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 14.2, we considered several transient noise pulse models including a template-based method, an AR model-based method and a hidden Markov model. In Sections 14.2 and 14.3, these models were applied to the detection and removal of noise pulses.

Bibliography

Godsill S.J. (1993), The Restoration of Degraded Audio Signals. PhD Thesis, Cambridge University.

Vaseghi S.V. (1987), Algorithm for Restoration of Archived Gramophone Recordings. Ph.D. Thesis, Cambridge University.



15

Echo Cancellation

Echo is repetition of a waveform either due to reflections at the points of impedance mismatch, where the characteristics of the medium through which the wave propagates suddenly changes, or due to the acoustic feedback between the speaker and the microphone of a communication system.

In telecommunication, echo degrades the quality of service, and echo cancellation is an essential part of communication systems. The development of echo reduction began in the late 1950s and continues today as new integration of landline, wireless cellular networks, Internet, multiple-input multiple-output communication devices, teleconferencing and wifi 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 direct and/or indirect acoustic 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 trunk 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 cancelling and removing line echo from telephone and data telecommunication systems, and acoustic feedback echoes from microphone–loudspeaker systems.

15.1 Introduction: Acoustic and Hybrid Echo

Echo is the repetition of a signal back to the transmitter; either due to a coupling between the loudspeaker and microphone or due to a reflection of the transmitted signal from the points or surfaces where the characteristics of the medium through which the signal propagates changes significantly so as to impede the propagation of the signal in the original direction such that some of the signal energy is reflected back to the source.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

The echo phenomenon is usefully employed for detection, exploration and navigation purposes in electronic surveillance and radar imaging instruments such as in sonar, ultrasonic imaging, infrared imaging and radar and also by some animals such as bats and dolphins.

Modern telecommunication systems (Figure 15.1) connect a variety of voice-enabled terminals, such as fixed telephones, mobile phones, laptops etc via a variety of networks and relays including public switched telephone network (PSTN), satellites, cellular networks, voice over internet protocol (VoIP), wifi, etc. Echo can severely affect the quality and intelligibility of voice conversation in telephone, teleconference, VoIP or cabin communication systems.

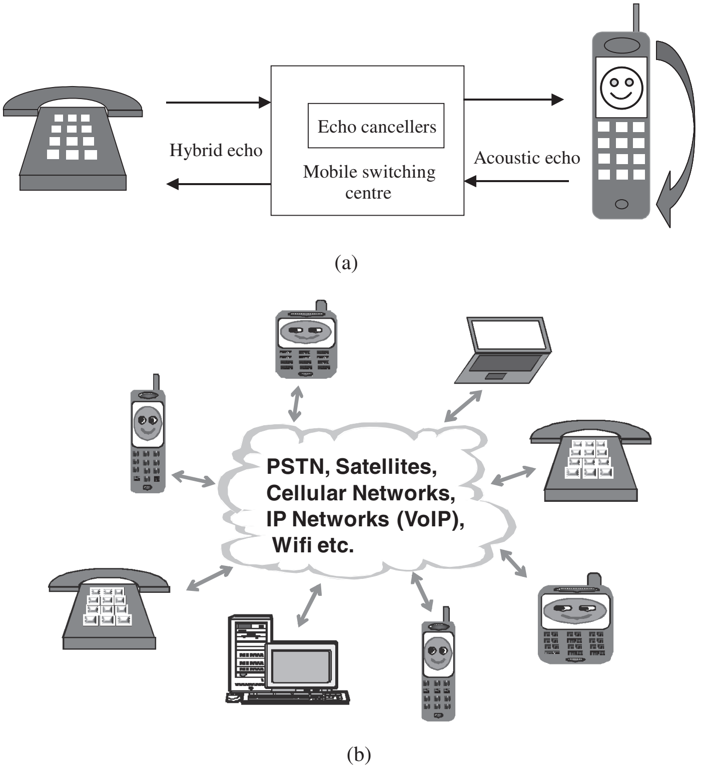


Figure 15.1 (a) Illustration of sources of echo in a mobile-to-landline system, (b) a modern communication network connects a variety of voice-enabled devices through a host of different telephone and IP networks.

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 msec 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 the 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 telecommunication systems such as conventional wire-line telephones, hands-free phones, cellular mobile (wireless) phones, teleconference systems and in-vehicle cabin communication systems.

There are two types of echo in a voice communication system (Figure 15.1(a)):

(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 subscriber's two-wire local line to the four-wire trunk line that connect switching centres or exchanges. 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 and two-wire lines were used for this purpose. Hence, at the exchange the four-wire trunk lines are converted to two-wire subscriber local lines using a four/two-wire hybrid bridge circuit. At the receiver due 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.

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 set, the speech signals are digitised, divided into frames, compressed, coded, packetised 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 ms to 100 ms, resulting in a total round-trip delay of 160–200 ms for any echo signals. In addition there is the signal propagation delay through the communication gateways and networks as explained in the next section. Delay of this magnitude will make any appreciable 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.

15.2 Echo Return Time: The Sources of Delay in Communication Networks

The delay, also known as latency, of an echo is the round-trip time taken for the signal to arrive back at the source.

In voice communication, end-to-end (i.e. one way) propagation delay is the sum of delays required for the voice of the speaker to be compressed, packetised, transmitted and propagated through different network devices and network links (including a combination of PSTN, Satellite and VoIP) to reach the listener. In general the end-to-end delay may be expressed as

$$\begin{aligned} Delay = {} & coder\ delay + packetisation\ delay + packet\ transmission\ (serialisation)\ delay \\ & + propagation\ delay + network\ (gateway)\ delay + decoder\ delay \end{aligned}$$

Round-trip delay of an echo is the propagation time from the transmitter to the receiver and then back to the transmitter.

Generally, voice communication operators regard an echo delay of up to 150 ms acceptable and a delay of above 400 ms seriously degrading to voice communication. Between 150 to 400 ms the echo may be tolerated.

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 more than, 600 ms, and echoes of this latency can become disruptive. For this reason the employment of echo cancellers in mobile switching centres and satellite networks is mandatory.

The magnitude and type of delays depends on the communication network and protocols, e.g. fixed switched network connection, asynchronous transfer mode (ATM), Voice over IP (VoIP), etc. There are two distinct types of delay in voice communication networks, called fixed and variable delays as explained next:

(1) *Fixed delay* components arise due to the time needed for signal processing (coding/decoding) modules, packetisation, serialisation and communication protocols and the finite propagation velocity of electromagnetic waves; these contribute directly to the overall delay on the connection. When the communication route is also fixed (e.g. switched networks) then the overall delay remains constant during a communication session.
(2) *Variable delays* (or jitters). Modern communication networks increasingly use packet transmission over internet where different packets of the same communication session may take different routes and experience different delays. Delay jitters arise from queuing delays in the egress trunk buffers on the serial port connected to the network. These buffers create variable delays, called delay jitter which are handled through the de-jitter buffer at the receiving end.

The sources of delay in telephone and VoIP networks are as follows.

15.2.1 Transmission link (electromagnetic wave propagation) delay

- *Transmission line delay*: this is caused by the finite propagation velocity of electromagnetic waves which, depending on the dielectric constant of the transmission medium, is between 0.66 to 1.0 times speed of light (300,000 km/sec), the latter happens in the free space vaccum. The transmission link delay is about 0.8–0.6 msec per 100 miles for copper coaxial cable or fiber optics and 0.54 msec per 100 miles (speed of light) on air for wireless.
- *Satellite link delay*: The propagation time of radio signals at velocity of light in free space to a geostationary satellite at a distance of about 35,786 km is about 120 ms which would then need to be relayed down to the destination. The satellite propagation delay is between for 250–300 msec; multiple hops can yield longer delays.

15.2.2 Speech coding/decoding delay

- *Speech coder delay*: 1–40 msec, includes signal segmentation delay and signal processing time for speech compression algorithm. For example, algebraic code excited linear prediction (ACELP) algorithms analyse and compress 10 ms blocks of PCM samples. Coder delay may include an additional look-ahead delay (typically 5 ms) where the signal in the next speech block is also used in the compression of the current block. However, this look-ahead delay is only a constant 5 ms addition to the overall delay.
- *Speech decoder delay*: typically less than 10 msec.

15.2.3 Network processing delay

- *Packetisation delay*: is the duration of packet, i.e. the time taken to fill a packet payload with encoded/compressed speech. This delay is a function of the sample block size required by the voice coder and the number of blocks placed in a single frame. For example with a block size of 10 ms, three blocks may be used in a frame. Note that there can be some pipelining of the delay as some of the packetisation delay may be overlapped with the speech block coder delay. Voice network developers aim for a packetisation delay of no more than 30 ms.

- *Transmission (serialisation or offloading) delay*: note this is not the propagation time over the communication link, this is the time it takes to clock on all the bits in a packet onto the network interface i.e. to place them on transmission line.
- *Processing delay*: In packet switching networks, processing delay is the time taken for the routers/gateways to process the packet header for sending it to the correct destination and to check (and correct) for bit errors in the packet that may have occurred during transmission. Processing delays in high-speed routers are small; typically on the order of microseconds or less.
- *Queuing delay* is the sum of the delays encountered by packets in transmission from source to destination. Queuing delay depends on bandwidth and demand and increases with congestion at times high network demand.
- *Voice over IP gateway node delay*: 50–100 msec and inter-process hand-offs delay: about 10 msec at each end.

15.2.4 De-Jitter delay

The jitter from all the variable delays must be removed before the signal leaves the network. This may be accomplished with a de-jitter buffer at the receiving-end router/gateway. The de-jitter buffer transforms the variable delay of buffer into a fixed delay. It holds the first sample received for a period of time before it plays it out. This holding period is known as the initial play out delay.

15.2.5 Acoustic echo delay

Acoustic echo delay can be longer that network echo delay. 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 meters/sec at a room temperature of 25 °C. Hence the time taken for sound to travel one meter will be about 2.94 msec. A distance of 10 meters from speaker to microphone will take about 29.4 msec and to this must be added the delay for coding and transmission through the communication network as described above.

15.3 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 or switching centre by a two-wire twisted line, called the subscriber's landline, which serves to receive and transmit signals, hence 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 not a 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 msec) 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 15.2. The device that connects the two-wire subscriber's loop to the four-wire line is called a hybrid, and is shown in Figure 15.3. 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 and hence

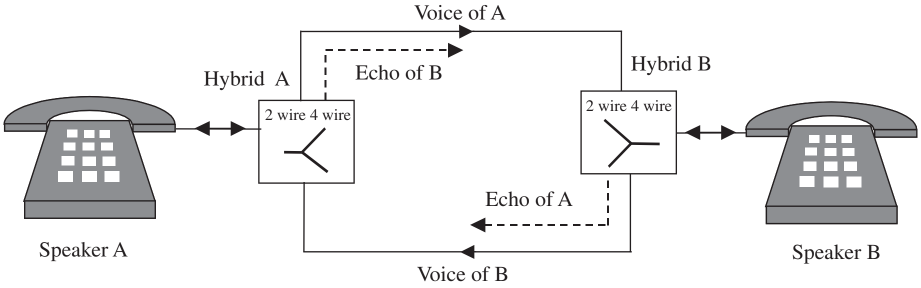


Figure 15.2 Illustration of a telephone call set up by connection of a two-wire subscriber's phone via hybrids to four-wire lines at the exchange. Note a two-wire twisted line between subscriber's telephone and exchange serves to transmit and receive signals, whereas between two exchanges separate two-wire twisted lines are used for transmission/reception in each direction.

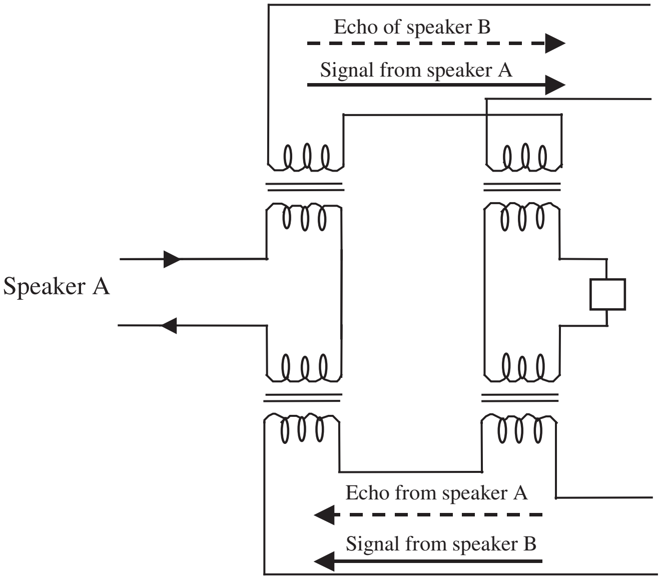


Figure 15.3 A two-wire to four-wire hybrid connection circuit.

no echo. However, each 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.

15.3.1 Echo Return Loss

The intensity of echo is measured in terms of the echo return loss (ERL) defined as the power ratio, in dB, of the transmitted signal to that of the returned echo as

$$\mathrm{ERL} = 10\log_{10}\left(\frac{\text{Transmitted Signal Power}}{\text{Echo Return Signal Power}}\right)\ \text{dB} \tag{15.1}$$

The higher the echo return loss 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 echo cancellation or equivalently it is the attenuation in echo measured in dB.

15.4 Hybrid (Telephone Line) 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 suppresser (Figure 15.4) is primarily a switch that lets the speech signal through during the speech-active periods and attenuates the line echo 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 speaker B to 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.

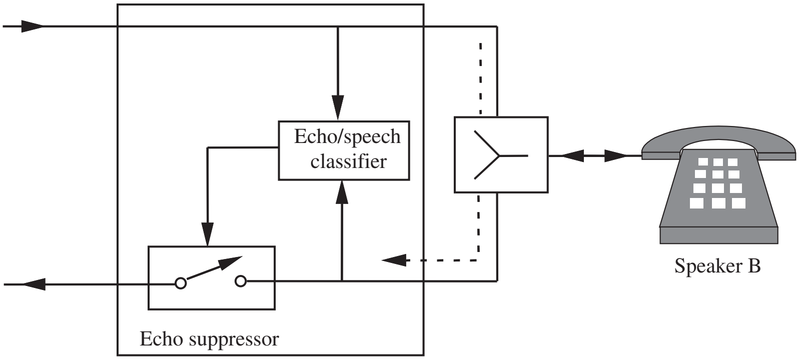


Figure 15.4 Block diagram illustration of an echo suppression system.

The performance of an echo suppresser depends on the accuracy of the echo/speech classification subsystem. The echo of speech often has a smaller amplitude level 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.

15.5 Adaptive Echo Cancellation

Echo cancellation was developed in the early 1960s by AT&T Bell Labs 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 15.5 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 4/2 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 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 (IIR) or a finite impulse response (FIR) 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.

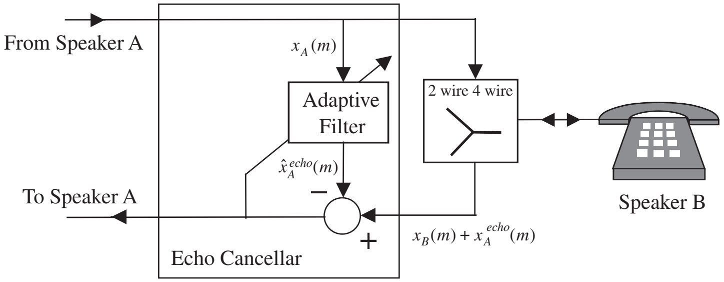


Figure 15.5 Block diagram illustration of an adaptive echo cancellation system.

Assuming that the signal on the line from speaker B to speaker A, $y_{\mathrm{B}}(m)$, is composed of the speech of speaker B, $x_{\mathrm{B}}(m)$, plus the echo of speaker A, $x_{\mathrm{A}}^{\mathrm{echo}}(m)$, we have

$$y_{\mathrm{B}}(m) = x_{\mathrm{B}}(m) + x_{\mathrm{A}}^{\mathrm{echo}}(m) \tag{15.2}$$

In practice, speech and echo signals are not simultaneously present on a phone line unless both speakers are speaking simultaneously. This, as pointed out shortly, can be used to simplify the adaptation process. Assuming that the truncated impulse response of the echo path can be modelled by an FIR filter, the output estimate of the synthesised echo signal can be expressed as

$$\hat{x}_{\mathrm{A}}^{\mathrm{echo}}(m) = \sum_{k=0}^{P} h_k(m) x_{\mathrm{A}}(m-k) \tag{15.3}$$

where $h_k(m)$ are the time-varying coefficients of an adaptive FIR filter model of the echo path and $\hat{x}_{\mathrm{A}}^{\mathrm{echo}}(m)$ 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

$$\begin{aligned} e(m) &= y_{\mathrm{B}}(m) - \hat{x}_{\mathrm{A}}^{\mathrm{echo}}(m) \\ &= x_{\mathrm{B}}(m) + x_{\mathrm{A}}^{\mathrm{echo}}(m) - \sum_{k=0}^{P} h_k(m) x_{\mathrm{A}}(m-k) \end{aligned} \tag{15.4}$$

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

$$\begin{aligned} e(m) &= \tilde{x}_{\mathrm{A}}^{\mathrm{echo}}(m) = x_{\mathrm{A}}^{\mathrm{echo}}(m) - \hat{x}_{\mathrm{A}}^{\mathrm{echo}}(m) \\ &= x_{\mathrm{A}}^{\mathrm{echo}}(m) - \sum_{k=0}^{P} h_k(m) x_{\mathrm{A}}(m-k) \end{aligned} \tag{15.5}$$

where $\tilde{x}_{\mathrm{A}}^{\mathrm{echo}}(m)$ is the residual echo. An echo canceller using an adaptive FIR filter is illustrated in Figure 15.6. 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.

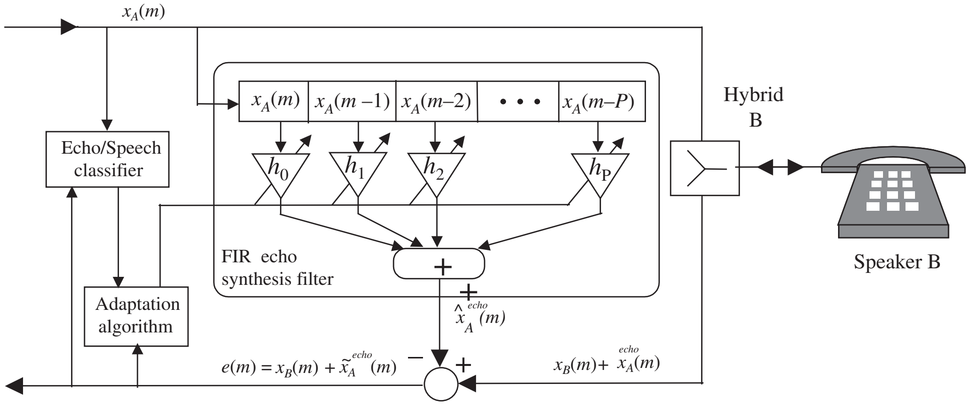


Figure 15.6 Illustration of an echo canceller using an adaptive FIR filter and incorporating an echo/speech classifier.

15.5.1 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 signals $x_{\mathrm{A}}(m)$ and $x_{\mathrm{B}}(m)$ are uncorrelated, the energy on the telephone line from B to A is minimised when the echo canceller output $\hat{x}_{\mathrm{A}}^{\mathrm{echo}}(m)$ is equal to the echo $x_{\mathrm{A}}^{\mathrm{echo}}(m)$ on the line. The echo canceller coefficients may be adapted using one of the variants of the recursive least square error (RLS) or the least mean squared error (LMS) 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

$$\boldsymbol{h}(m) = \boldsymbol{h}(m-1) + \mu \frac{e(m)}{\boldsymbol{x}(m)_{\mathrm{A}}^{\mathrm{T}} \boldsymbol{x}_{\mathrm{A}}(m)} \boldsymbol{x}_{\mathrm{A}}(m) \tag{15.6}$$

where $\boldsymbol{x}_{\mathrm{A}}^{\mathrm{T}}(m) = [x_{\mathrm{A}}(m), \ldots, x_{\mathrm{A}}(m-P)]$ and $\boldsymbol{h}^{\mathrm{T}}(m) = [h_0(m), \ldots, h_P(m)]$ 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 $\boldsymbol{x}(m)_{\mathrm{A}}^{\mathrm{T}} \boldsymbol{x}_{\mathrm{A}}(m)$ 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.

15.5.2 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) *Non-stationary 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 non-stationary 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 talk*. In a telephone conversation, usually the speakers do not talk 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 15.6. 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 algorithm are its relatively low memory and computational requirements and its ease of implementation and monitoring. The main drawback of the LMS algorithm 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 recursive least square (RLS) error methods have a faster convergence rate, are less sensitive to the eigenvalue spread of the signal and have a better minimum mean square error performance. With the increasing availability of low-cost high-speed dedicated DSP processors, implementation of higher performance and computationally intensive echo cancellers based on RLS are now feasible.

15.5.3 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 two-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 subscriber's 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 kilobits per second (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. Using advanced signal processing and modulation schemes methods such as ADSL can achieve a 10 megabits per second data rate over 240 MHz bandwidth of subscriber's twisted wire line.

Figure 15.7 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-end 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 echo 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.

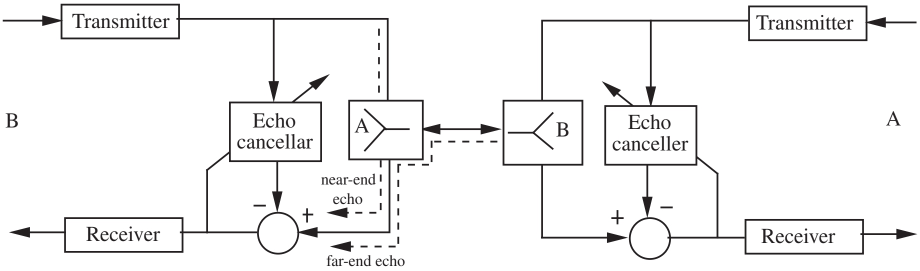


Figure 15.7 Echo cancellation in digital modems using two-wire subscriber's loop.

15.6 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. The echo feedback path may include a direct speaker-microphone path and a number of indirect paths along which the signal from the speaker is reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through different paths before it is picked up by the microphone.

If the echo 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.

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 non-linearity of the electronic system.

There are a number of methods for removing acoustic feedback. One method for alleviating the effects of acoustic feedback and the 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 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 by 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.

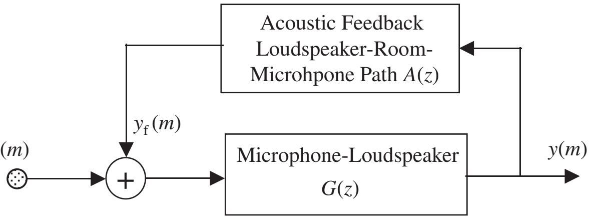


Figure 15.8 Configuration of a feedback model for a microphone–loudspeaker–room system.

The most effective method of acoustic feedback removal is the use of an adaptive feedback cancellation system. Figure 15.8 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

$$H(z) = \frac{G(z)}{1 - G(z)A(z)} \tag{15.7}$$

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 multi-path reflections of a room environment. Assuming that the microphone–loudspeaker combination has a flat frequency response with a gain of G, Equation (15.7) can be simplified to

$$H(z) = \frac{G}{1 - GA(z)} \tag{15.8}$$

Note that in Equations (15.7) and (15.8), 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 (15.8) is given by the following difference equation:

$$y(m) = \sum_{k=0}^{P} a_k(m)y(m-k) + Gx(m) \tag{15.9}$$

where $a_k(m)$ are the coefficients 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 15.9 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 $y_f(m)$:

$$y(m) = x(m) + y_f(m) \tag{15.10}$$

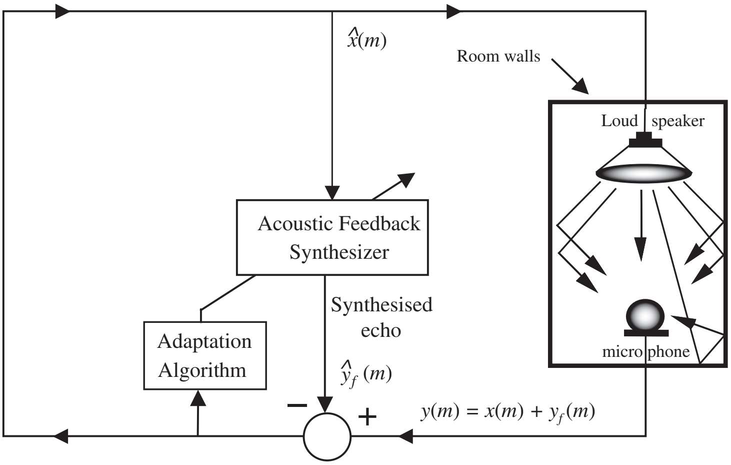


Figure 15.9 Illustration of adaptive acoustic feedback cancellation in a conference room environment.

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

$$\hat{y}_f(m) = \sum_{k=0}^{P} a_k(m)y(m-k) \tag{15.11}$$

The filter coefficients are adapted to minimise the energy of an error signal defined as

$$e(m) = x(m) + y_f(m) - \hat{y}_f(m) \tag{15.12}$$

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 of the order of a few thousand coefficients may be required. The stable and speedy adaptation of filters of such length presents a difficult problem. Secondly, the characteristics of an acoustic echo path are more non-stationary compared with those 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 a non-linear process, 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, Figure 15.10. The maximum usable gain of a hearing aid system is limited by the acoustic feedback between the microphone and the speaker. 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.

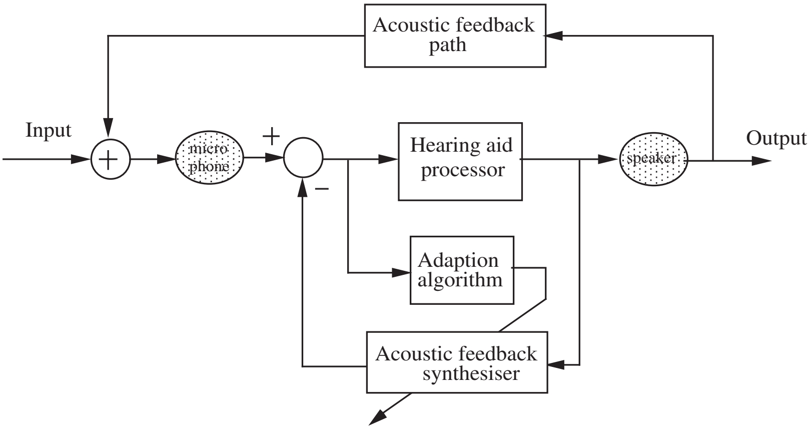


Figure 15.10 Configuration of an acoustic feedback canceller incorporated in a hearing aid system.

15.7 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 15.11. The sub-band analyser

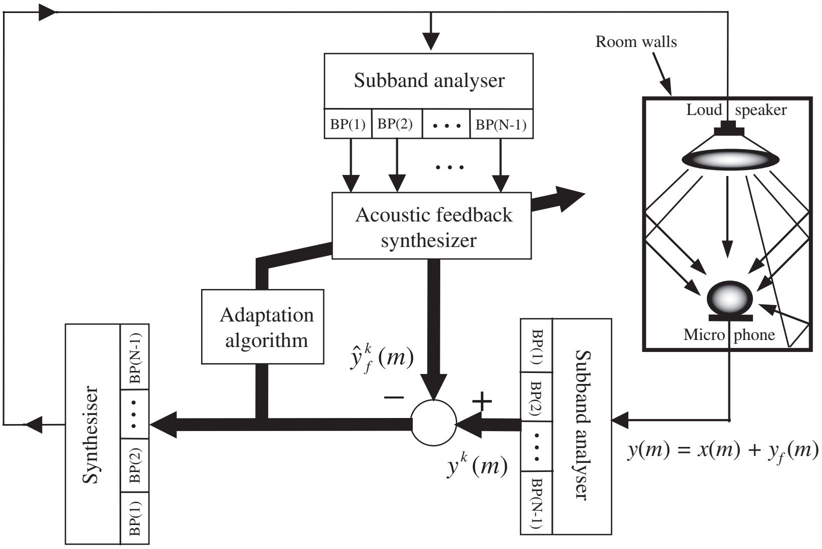


Figure 15.11 Configuration of a sub-band acoustic echo cancellation system.

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.

15.8 Echo Cancellation with Linear Prediction Pre-whitening

Adaptive echo cancellation systems work better (i.e. converge better) if the input and the reference signals are uncorrelated white noise processes. Speech signals are highly correlated but can be pre-whitened by first modelling the speech with a linear prediction model and then using an inverse linear predictor for whitening the signals as illustrated in Figure 15.12. Linear prediction models and pre-whitening filters

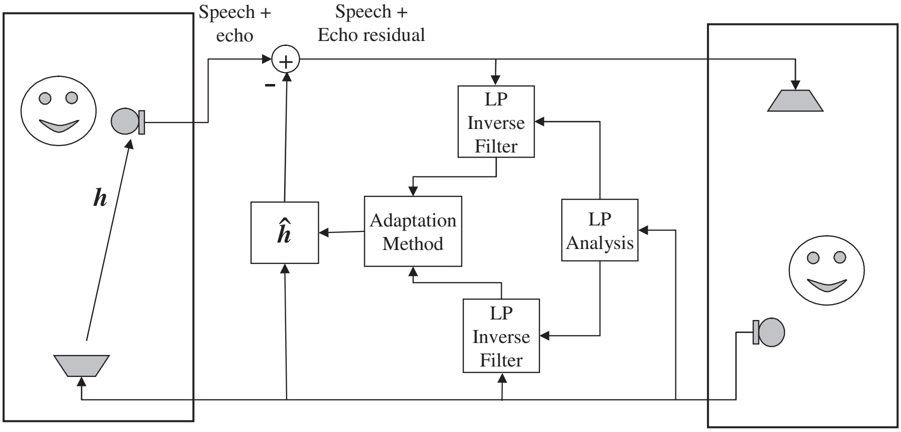


Figure 15.12 An acoustic echo cancellation system incorporating inverse linear prediction filters as pre-whitening filters.

are described in detail in Chapter 8. The pre-whitened input to adaptive filter, i.e. pre-whitened incoming signal, is given by

$$e(m) = x(m) - \sum_{k=1}^{P} a_k x(m-k) \tag{15.13}$$

A similar equation can be used to pre-whiten the adaptive echo canceller's reference signal as shown in Figure 15.12.

Note that for the purpose of synthesis of the echo the input to the filter $\widehat{\boldsymbol{h}}$ is the non-whitened speech signal, whereas for the purpose of the adaptation of the filter coefficients $\widehat{\boldsymbol{h}}$ the whitened speech and whitened reference signals are used.

The process of pre-whitening the input and reference signals of the adaptive filter can substantially improve the performance of echo cancellation systems.

15.9 Multi-Input Multi-Output 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 an acoustic feedback path is 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 may be modelled by an FIR filter $\boldsymbol{h}_{ij}$. The truncated impulse response of each acoustic path from human speaker i to microphone j is modelled by an FIR filter $\boldsymbol{g}_{ij}$.

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 later.

15.9.1 Stereophonic Echo Cancellation Systems

Figure 15.13 shows the configuration of an 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 multi-path reflections of 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

$$x_r(m) = \boldsymbol{g}_r^T(m)\boldsymbol{s}(m) \text{ and } x_l(m) = \boldsymbol{g}_l^T(m)\boldsymbol{s}(m) \tag{15.14}$$

where $\boldsymbol{g}_r$ and $\boldsymbol{g}_l$ are the truncated room impulse responses from the source speaker to the right and left microphones respectively.

In Figure 15.13, the truncated impulse responses of the acoustic feedback paths from the right and left loudspeakers to the right microphone are denoted as to $\boldsymbol{h}_{rr}(m)$ and $\boldsymbol{h}_{lr}(m)$ respectively and combined as $\boldsymbol{h}_r^T(m) = [\boldsymbol{h}_{rr}^T(m),\ \boldsymbol{h}_{lr}^T(m)]$. The signals from the right and left loudspeakers, $\boldsymbol{x}_r(m)$ and $\boldsymbol{x}_l(m)$, may be combined as $\boldsymbol{x}^T(m) = [\boldsymbol{x}_r^T(m),\ \boldsymbol{x}_l^T(m)]$. There exist similar paths from loudspeakers to each microphone

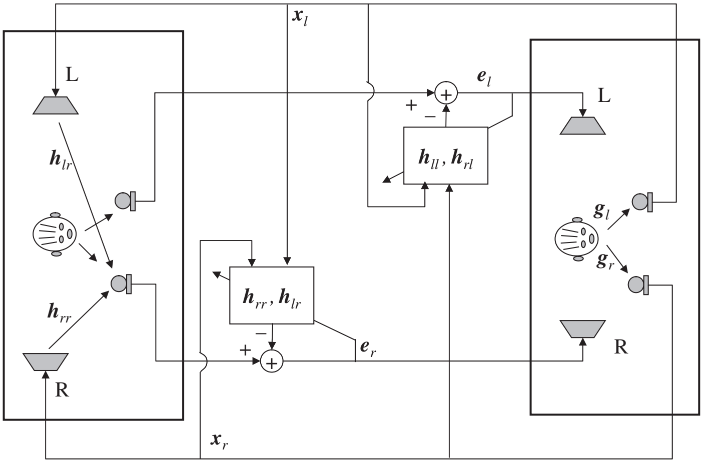


Figure 15.13 Illustration of the feedback signals and adaptive cancellation of acoustic feedbacks in a stereophonic echo-cancellation system.

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

$$\hat{x}_{\text{echo},\boldsymbol{r}}(m) = \underbrace{\hat{\boldsymbol{h}}_{\boldsymbol{rr}}^{\boldsymbol{T}}(m-1)\boldsymbol{x}_{\boldsymbol{r}}(m)}_{\substack{\text{Synthesised echo from right}\\ \text{loudspeaker to right microphone}}} + \underbrace{\hat{\boldsymbol{h}}_{\boldsymbol{lr}}^{\boldsymbol{T}}(m-1)\boldsymbol{x}_{\boldsymbol{l}}(m)}_{\substack{\text{Synthesised echo from left}\\ \text{loudspeaker to right microphone}}} \tag{15.15}$$

$$= \hat{\boldsymbol{h}}_{\boldsymbol{r}}^{\boldsymbol{T}}(m-1)\boldsymbol{x}(m)$$

The error signal composed of speech and the residue echo is given by

$$e_r(m) = y(m) - \hat{x}_{\text{echo},\,\boldsymbol{r}}(m) \tag{15.16}$$

where $y(m)$ is the signal plus echo and noise picked up by the left microphone. The NLMS adaptation method for FIR model of stereophonic echo paths is given by

$$\hat{\boldsymbol{h}}_{\boldsymbol{r}}(m) = \hat{\boldsymbol{h}}_{\boldsymbol{r}}(m-1) + \mu \frac{e(m)}{\boldsymbol{x}(m)^T\boldsymbol{x}(m)}\boldsymbol{x}(m) \tag{15.17}$$

Similar equations describing the echo and adaptive echo cancellation can be written for the right microphones.

15.9.2 Non-uniqueness 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 $\boldsymbol{x}(m) = [\boldsymbol{x}_r(m)\ \boldsymbol{x}_l(m)]$ and the channels to be identified, for example from right and left loudspeakers to say the right microphone, are $\boldsymbol{h}_r(m) = [\boldsymbol{h}_{rr}(m)\ \boldsymbol{h}_{lr}(m)]$. The Wiener solution to this echo path estimation problem is given by

$$\hat{\boldsymbol{h}}_r = \boldsymbol{R}_{\boldsymbol{xx}}^{-1}\boldsymbol{r}_{xy} \tag{15.18}$$

where $\boldsymbol{R}_{xx}$ is the autocorrelation matrix of $\boldsymbol{x}(m) = [\boldsymbol{x}_r(m)\ \boldsymbol{x}_l(m)]$. The problem is that due to the high correlation of $\boldsymbol{x}_r(m)$ and $\boldsymbol{x}_l(m)$ the autocorrelation matrix $\boldsymbol{R}_{xx}$ is not a full rank matrix and hence the solution is not unique.

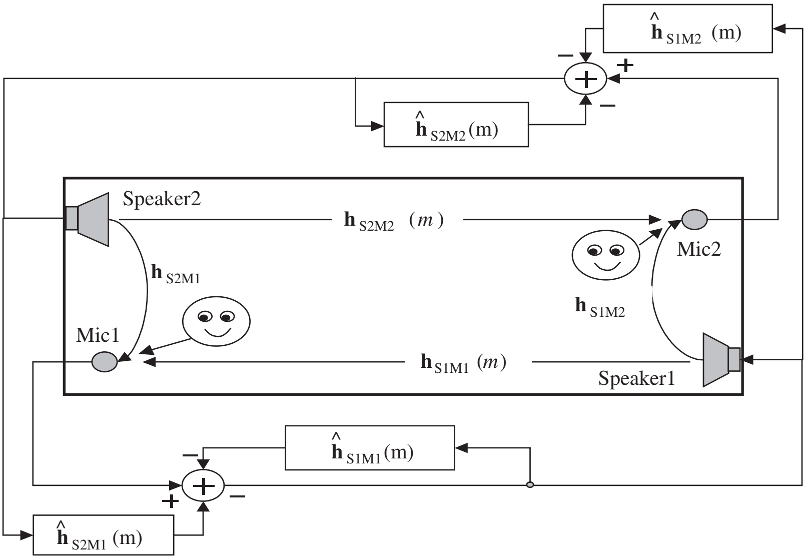
The non-uniqueness problem can also be explained in the frequency domain by considering the sum of the feedbacks from the loudspeakers into say the right microphone $X_{\text{echo},r}(f)$:

$$\begin{aligned} X_{\text{echo},r}(f) &= H_{lr}(f)G_l(f)S(f) + H_{rr}(f)G_r(f)S(f) \\ &= [H_{lr}(f)G_l(f) + H_{rr}(f)G_r(f)]S(f) \end{aligned} \tag{15.19}$$

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 (15.19). A solution to this problem is to de-correlate 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 (1998).

15.9.3 MIMO In-Cabin Communication Systems

MIMO systems have application for in-car cabin communication systems (CCS) for multipurpose and large vehicles. The problem in CCS systems is background noise reduction and acoustic feedback cancellation. Figure 15.14 illustrates the 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 reach the microphone. The synthesised feedback is subtracted from the signal received by each microphone in order to cancel feedback echo signals.



$\hat{\mathbf{h}}_{S1M2}$ (m)
−
+
−
$\hat{\mathbf{h}}_{S2M2}$(m)
Speaker2
$\mathbf{h}_{S2M2}$ (m)
Mic2
$\mathbf{h}_{S2M1}$
$\mathbf{h}_{S1M2}$
Mic1
$\mathbf{h}_{S1M1}$ (m)
Speaker1
$\hat{\mathbf{h}}_{S1M1}$(m)
−
+
−
$\hat{\mathbf{h}}_{S2M1}$ (m)

Figure 15.14 A block diagram illustration of a MIMO echo-cancellation in car 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 17, form a beam where any signal arriving from directions outside the beam is attenuated. This would help to reduce the feedback and can be used in combination with echo cancellers.

15.10 Summary

Telephone line echo and acoustic feedback echo affect the functioning of telecommunication 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 Sections 15.3 and 15.4, hybrid telephone line echo and echo suppression were considered. In Section 15.5 adaptive line echo cancellation was considered where a synthesised replica of the echo is and subtracted from it. For adaptation of an echo canceller, the LMS or the RLS adaptation methods can be used. The RLS methods provide a faster convergence rate and better overall performance at the cost of higher computational complexity.

In Section 15.6, we considered the problem of 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 15.7 alleviates these problems. In Section 15.8 echo cancellation with pre-whitening via inverse linear prediction filters is considered. This method accelerates the convergence rate of adaptation.

For stereophonic and MIMO systems the problem of acoustic echo cancellation, introduced in Section 15.9, 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**, 3: 1611–1617.

Benesty J., Morgan D.R. and 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*, **6**(2):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. Am.*, **78** (5): 1508–1518.

Flanagan J.L. *et al*. (1991) Autodirective Microphone systems. *Acoustica* **73**: 58–71.

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*. Proc. Int. Conf. Acoustics, Speech and Signal Processing, **ICASSP–91**(3): 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**: 1403–1406.

Hansler E. (1992) The Hands-Free Telephone Problem – An Annotated Bibliography. *Signal Processing*, **27**: 259–271.

Hart J.E., Naylor P.A. and Tanrikulu O. (1993) Polyphase Allpass IIR Structures for Sub-band Acoustic Echo Cancellation. EuroSpeech-**93**(3,): 1813–1816.

Hua Ye and Bo-Xia Wu (1991) A New Double-Talk Detection Algorithm Based on the Orthogonality Theorem. *IEEE Trans. Communications*, **39**(11): 1542–1545.

Kellermann W. (1988) Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes. *IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing*, **ICASSP-88**: 2570–2573.

Knappe M.E. (1992) Acoustic Echo Cancellation: Performance and Structures. M. Eng. Thesis, Carleton University, Ottawa, Canada.

Martin R. and Altenhoner J. (1995) Coupled Adaptive Filters for Acoustic Echo Control and Noise Reduction. *IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing*, **ICASSP-95**(5): 3043–3046.

McCaslin S.R., Hemkumar N. and Redheendran B. (1997) Double-Talk Detector for Echo Canceller. US Patent No. 5631900, May 20.

Oslen H.F. (1964) *Acoustical Engineering*. D. Van Nostrand, Toronto.

Schroeder M.R. (1964) Improvement of Acoustic-Feedback Stability by Frequency Shifting. *J. Acoust. Soc. Amer*., **36**: 1718–1724.

Silverman H. 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 Chambers J.A. (1995) Finite-Precision Design and Implementation of All-Pass Polyphase Networks for Echo Cancellation in Sub-bands. *IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing*, **ICASSP-95**(5): 3039–3042.

Vaidyanathan P.P. (1993) *Multirate Systems and Filter Banks*. Prentice-Hall.

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.

16

Channel Equalisation and Blind Deconvolution

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, in seismic data analysis, etc.

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.

16.1 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 equalisation. Figure 16.1 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

$$y(m) = h[x(m)] + n(m) \tag{16.1}$$

where the function $h[\cdot]$ is the channel distortion. In general, the channel response may be time-varying and non-linear. In this chapter, it is assumed that the effects of a channel can be modelled using a stationary,

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

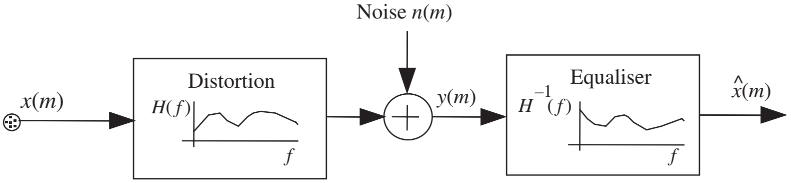


Figure 16.1 Illustration of a channel distortion model followed by an equaliser.

or a slowly time-varying, linear transversal filter. For a linear transversal filter model of the channel, Equation (16.1) becomes

$$y(m) = \sum_{k=0}^{P-1} h_k(m)x(m-k) + n(m) \tag{16.2}$$

where $h_k(m)$ are the coefficients 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 (16.2) becomes

$$Y(f) = X(f)H(f) + N(f) \tag{16.3}$$

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 (16.3) yields

$$\ln|Y(f)| = \ln|X(f)| + \ln|H(f)| \tag{16.4}$$

From Equation (16.4), in the log-frequency domain the effect of channel distortion is the addition of a 'tilt' term $\ln|H(f)|$ to the signal spectrum.

16.1.1 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

$$H(f)H^{\text{inv}}(f) = 1 \tag{16.5}$$

In Equation (16.5) $H^{\text{inv}}(f)$ 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 (16.5) is given by the z-transform relation

$$H(z)H^{\text{inv}}(z) = z^{-N} \tag{16.6}$$

for some value of the delay N that makes the channel inversion process causal. Taking the inverse Fourier transform of Equation (16.5), we have the following convolutional relation between the impulse responses of the channel $\{h_k\}$ and the ideal inverse channel response $\{h_k^{\text{inv}}\}$:

$$\sum_k h_k^{\text{inv}} h_{i-k} = \delta(i) \tag{16.7}$$

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

$$\begin{aligned}\hat{x}(m) &= \sum_k h_k^{\text{inv}} y(m-k) \\ &= \sum_k h_k^{\text{inv}} \sum_j h_j x(m-k-j) \\ &= \sum_i x(m-i) \sum_k h_k^{\text{inv}} h_{i-k}\end{aligned} \tag{16.8}$$

The last line of Equation (16.8) 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 (16.7) in Equation (16.8) yields

$$\hat{x}(m) = \sum_i \delta(i) x(m-i) = x(m) \tag{16.9}$$

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 non-invertible. 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.

16.1.2 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:

$$\begin{aligned}v(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k)\end{aligned} \tag{16.10}$$

where $\hat{h}_k^{\text{inv}}$ 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

$$x(m) = \sum_{k=0}^{P-1} h_k^{\text{inv}} y(m-k) \tag{16.11}$$

Substitution of Equation (16.11) in Equation (16.10) yields

$$\begin{aligned}v(m) &= \sum_{k=0}^{P-1} h_k^{\text{inv}} y(m-k) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \\ &= \sum_{k=0}^{P-1} \tilde{h}_k^{\text{inv}} y(m-k)\end{aligned} \tag{16.12}$$

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 $\tilde{h}_k^{\text{inv}}$ 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.

16.1.3 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, non-linear, and may also be non-invertible. 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 16.2(a). The obvious drawback of using training periods for channel equalisation 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 16.2(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.

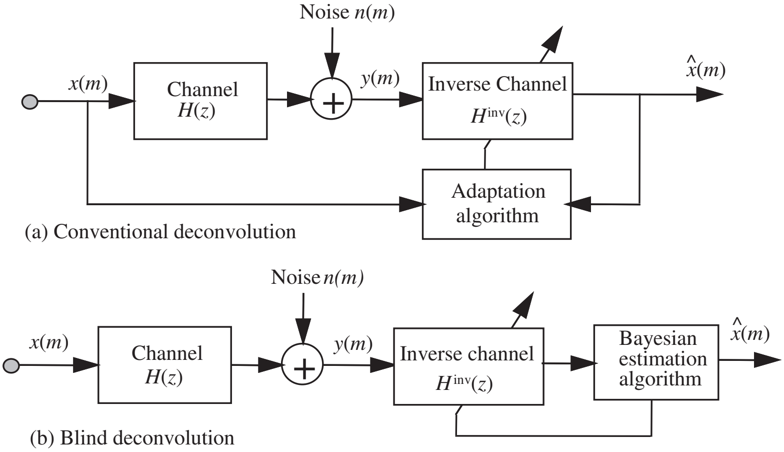


Figure 16.2 A comparative illustration of (a) a conventional equaliser with access to channel input and output, and (b) a blind equaliser.

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:

(1) *Channel identification*. The general form of a channel estimator can be expressed as

$$\hat{\boldsymbol{h}} = \psi(\boldsymbol{y}, \mathcal{M}_x, \mathcal{M}_h) \tag{16.13}$$

where ψ is the channel estimator, the vector $\hat{\boldsymbol{h}}$ is an estimate of the channel response, $\boldsymbol{y}$ 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:

(a) The distribution of the channel input signal: for example, in decision-directed channel equalisation, described in Section 16.5, 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 equaliser adaptation process.
(b) 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 16.3.1 to estimate a stationary channel from the long-time averages of the channel output.
(c) The stationary, or time-varying characteristics of the input signal process and the channel: in Section 16.3.1, a method is described for the recovery of a non-stationary signal convolved with the impulse response of a stationary channel.

(2) *Channel equalisation*. Assuming that the channel is invertible, the channel input signal $x(m)$ can be recovered using an inverse channel filter as

$$\hat{x}(m)=\sum_{k=0}^{P-1}\hat{h}_k^{\text{inv}}y(m-k) \tag{16.14}$$

In the frequency domain, Equation (16.14) becomes

$$\hat{X}(f)=\hat{H}^{\text{inv}}(f)Y(f) \tag{16.15}$$

In practice, perfect recovery of the channel input may not be possible, either because the channel is non-invertible or because the output is observed in noise. A channel is non-invertible if:

(a) 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.
(b) 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 16.3 illustrates the frequency response of a channel that has one invertible and two non-invertible regions. In the non-invertible 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.

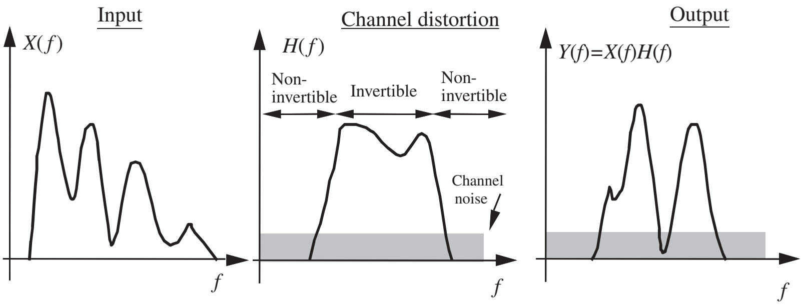


Figure 16.3 Illustration of the invertible and noninvertible regions of a channel.

16.1.4 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 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 16.4 illustrates examples of maximum-phase and minimum-phase fourth-order FIR filters.

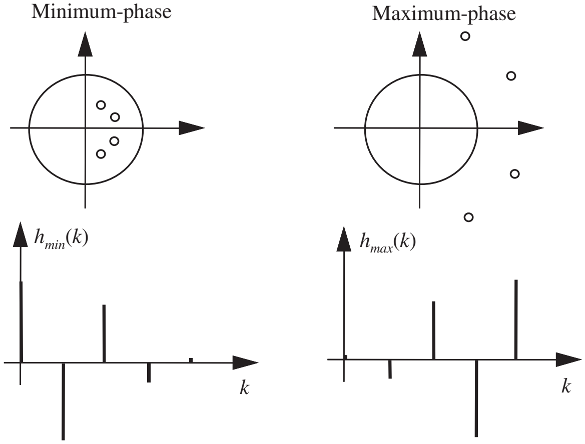


Figure 16.4 Illustration of the zero diagram and impulse response of fourth order maximum-phase and minimum-phase FIR filters.

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. correlation, covariance or 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 non-linear estimate of the desired signal to direct the adaptation of a channel equaliser as in Section 16.5, or we can use the higher-order statistics as in Section 16.6.

16.1.5 Wiener Equaliser

In this section, we consider the least squared 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

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{16.16}$$

where $\hat{h}_k^{\text{inv}}$ is an FIR Wiener filter estimate of the inverse channel impulse response. The equalisation error signal $v(m)$ is defined as

$$v(m) = x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{16.17}$$

The Wiener equaliser with input $y(m)$ and desired output $x(m)$ is obtained from Equation (8.10) in Chapter 6 as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{xy} \tag{16.18}$$

where $\boldsymbol{R_{yy}}$ is the $P \times P$ autocorrelation matrix of the channel output, and $\boldsymbol{r}_{xy}$ is the P-dimensional cross-correlation vector of the channel input and output signals. A more expressive form of Equation (16.18) can be obtained by writing the noisy channel output signal in vector equation form as

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n} \tag{16.19}$$

where $\boldsymbol{y}$ is an N-sample channel output vector, $\boldsymbol{x}$ is an $N+P$-sample channel input vector including the P initial samples, $\boldsymbol{H}$ is an $N \times (N+P)$ channel distortion matrix whose elements are composed of the coefficients of the channel filter, and $\boldsymbol{n}$ is a noise vector. The autocorrelation matrix of the channel output can be obtained from Equation (16.19) as

$$\boldsymbol{R_{yy}} = \mathcal{E}[\boldsymbol{yy}^{\text{T}}] = \boldsymbol{HR_{xx}H}^{\text{T}} + \boldsymbol{R_{nn}} \tag{16.20}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. The cross-correlation vector $\boldsymbol{r}_{xy}$ of the channel input and output signals becomes

$$\boldsymbol{r}_{xy} = \mathcal{E}[x\boldsymbol{y}] = \boldsymbol{Hr}_{xx} \tag{16.21}$$

Substitution of Equation (16.20) and (16.21) in (16.18) yields the Wiener equaliser as

$$\hat{\boldsymbol{h}}^{\text{inv}} = (\boldsymbol{HR_{xx}H}^{\text{T}} + \boldsymbol{R_{nn}})^{-1} \boldsymbol{Hr_{xx}} \tag{16.22}$$

The derivation of the Wiener equaliser in the frequency domain is as follows. The Fourier transform of the equaliser output is given by

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f) Y(f) \tag{16.23}$$

where $Y(f)$, 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

$$\begin{aligned} V(f) &= X(f) - \hat{X}(f) \\ &= X(f) - \hat{H}^{\text{inv}}(f) Y(f) \end{aligned} \tag{16.24}$$

As in Section 6.5 minimisation of the expectation of the squared magnitude of $V(f)$ results in the frequency Wiener equaliser given by

$$\begin{aligned} \hat{H}^{\text{inv}}(f) &= \frac{P_{XY}(f)}{P_{YY}(f)} \\ &= \frac{P_{XX}(f) H^*(f)}{P_{XX}(f) \left|H(f)\right|^2 + P_{NN}(f)} \end{aligned} \tag{16.25}$$

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 $H^{\text{inv}}(f) = H^{-1}(f)$.

16.2 Blind Equalisation Using 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 16.5, 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 as 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 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 (16.2), reproduced here for convenience

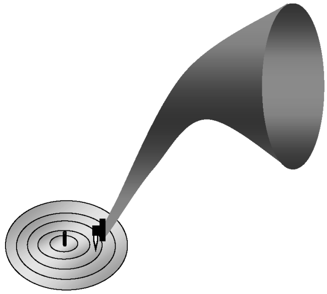


Figure 16.5 Illustration of the early acoustic recording process on a wax disc. Acoustic recordings were made by focusing the sound energy, through a horn via a sound box, diaphragm and stylus mechanism, onto a wax disc. The sound was distorted by reverberations of the horn.

$$y(m) = \sum_{k=0}^{P-1} h_k x(m-k) + n(m) \tag{16.26}$$

or in the frequency domain as

$$Y(f) = X(f)H(f) + N(f) \tag{16.27}$$

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 (16.27) with their complex conjugates, and taking the expectation, we obtain

$$\mathcal{E}[Y(f)Y^*(f)] = \mathcal{E}\left[\left(X(f)H(f)+N(f)\right)\left(X(f)H(f)+N(f)\right)^*\right] \tag{16.28}$$

Assuming the signal $X(f)$ and the noise $N(f)$ are uncorrelated Equation (16.28) becomes

$$P_{YY}(f) = P_{XX}(f)\,|H(f)|^2 + P_{NN}(f) \tag{16.29}$$

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 (16.29) an estimate of the spectrum of the channel response can be obtained as

$$|H(f)|^2 = \frac{P_{YY}(f) - P_{NN}(f)}{P_{XX}(f)} \tag{16.30}$$

In practice, Equation (16.30) is implemented using time-averaged estimates of the power spectra.

16.2.1 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 16.6. Through homomorphic transformation convolution becomes addition, and equalisation becomes subtraction.

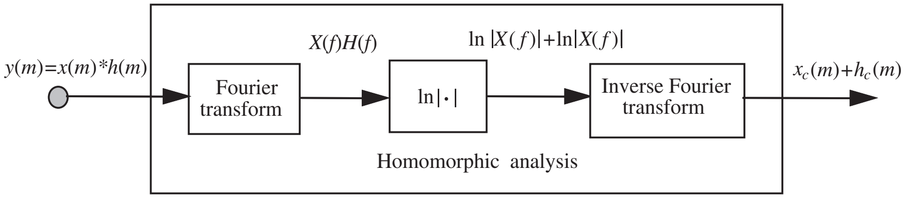


Figure 16.6 Illustration of homomorphic analysis in deconvolution.

Ignoring the additive noise term and transforming both sides of Equation (16.27) into log-spectral variables yields

$$\ln Y(f) = \ln X(f) + \ln H(f) \tag{16.31}$$

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 (16.31) yields

$$\mathcal{E}[\ln Y(f)] = \mathcal{E}[\ln X(f)] + \ln H(f) \tag{16.32}$$

In Equation (16.32), 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

$$\mathcal{E}[\ln X(f)] = \mathcal{E}[\ln|X(f)|] + j\mathcal{E}[\angle X(f)] \tag{16.33}$$

The first term on the right-hand side of Equation (16.33), $\mathcal{E}[\ln|X(f)|]$, is non-zero, 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 (16.32), the log-frequency spectrum of the channel can be estimated as

$$\ln H(f) = \mathcal{E}[\ln Y(f)] - \mathcal{E}[\ln X(f)] \tag{16.34}$$

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

$$y_i(m) \approx x_i(m) * h_i(m) \tag{16.35}$$

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 (16.35) yields

$$\ln Y_i(f) = \ln X_i(f) + \ln H_i(f) \tag{16.36}$$

Taking the time averages over N segments of the distorted signal record yields

$$\frac{1}{N}\sum_{i=0}^{N-1} \ln Y_i(f) = \frac{1}{N}\sum_{i=0}^{N-1} \ln X_i(f) + \frac{1}{N}\sum_{i=0}^{N-1} \ln H_i(f) \tag{16.37}$$

Estimation of the channel response from Equation (16.37) 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 that of the acoustic recording. From Equation (16.37), the estimate of the logarithm of the channel is given by

$$\ln \hat{H}(f) = \frac{1}{N}\sum_{i=0}^{N-1} \ln Y_i(f) - \frac{1}{N}\sum_{i=0}^{N-1} \ln X_i^{\mathcal{M}}(f) \tag{16.38}$$

where $X^{\mathcal{M}}(f)$ is the spectrum of a modern recording. The equaliser can then be defined as

$$\ln H^{\text{inv}}(f) = \begin{cases} -\ln \hat{H}(f), & 200\,\text{Hz} \leq f \leq 4000\,\text{Hz} \\ -40\,\text{dB}, & \text{otherwise} \end{cases} \tag{16.39}$$

In Equation (16.39), the inverse acoustic channel is implemented in the range between 200 and 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.

16.2.2 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

$$\ln Y_t(f) = \ln X_t(f) + \ln H_t(f) \tag{16.40}$$

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 (16.40), 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 narrowband 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

$$\ln \hat{X}_t(f) = \alpha \ln \hat{X}_{t-1}(f) + \ln Y_t(f) - \ln Y_{t-1}(f) \tag{16.41}$$

where the parameter α controls the bandwidth of the notch at zero frequency. Note that the filter bank also removes any dc component of the signal ln $X(f)$; for some applications, such as speech recognition, this is acceptable.

16.3 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

$$X(z) = E(z)A(z) \tag{16.42}$$

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

$$Y(z) = X(z)H(z) \tag{16.43}$$

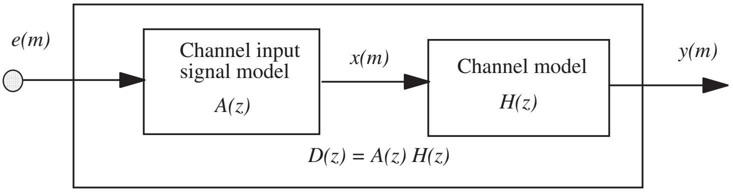


Figure 16.7 A distorted signal modelled as cascade of a signal model and a channel model.

Figure 16.7 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

$$\begin{aligned} Y(z) &= E(z)A(z)H(z) \\ &= E(z)D(z) \end{aligned} \tag{16.44}$$

where

$$D(z) = A(z)H(z) \tag{16.45}$$

The z-transfer function of the linear prediction models of the channel input signal and the channel can be expanded as

$$A(z) = \frac{G_1}{1 - \sum_{k=1}^{P} a_k z^{-k}} = \frac{G_1}{\prod_{k=1}^{P} (1 - \alpha_k z^{-1})} \tag{16.46}$$

$$H(z) = \frac{G_2}{1 - \sum_{k=1}^{Q} b_k z^{-k}} = \frac{G_2}{\prod_{k=1}^{Q} (1 - \beta_k z^{-1})} \tag{16.47}$$

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 (16.46) and (16.47) in Equation (16.45) yields the combined input-channel model as

$$D(z) = \frac{G}{1 - \sum_{k=1}^{P+Q} \mathrm{d}_k z^{-k}} = \frac{G}{\prod_{k=1}^{P+Q} (1 - \gamma_k z^{-1})} \tag{16.48}$$

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.

16.3.1 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 non-factorable 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 non-stationary 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

$$x(m) = a_1(m)x(m-1) + a_2(m)x(m-2) + G_1(m)e(m) \tag{16.49}$$

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 (16.49); these poles are the roots of the polynomial

$$1 - a_1(m)z^{-1} - a_2(m)z^{-2} = \left[1 - z^{-1}\alpha_1(m)\right]\left[1 - z^{-1}\alpha_2(m)\right] = 0 \tag{16.50}$$

Similarly, assume that the channel can be modelled by a second-order stationary linear predictive model as

$$y(m) = h_1 y(m-1) + h_2 y(m-2) + G_2 x(m) \tag{16.51}$$

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

$$1 - h_1 z^{-1} - h_2 z^{-2} = (1 - z^{-1}\beta_1)(1 - z^{-1}\beta_2) = 0 \tag{16.52}$$

The combined cascade of the two second-order models of Equations (16.49) and (16.51) can be written as a fourth-order linear predictive model with input $e(m)$ and output $y(m)$:

$$y(m) = d_1(m)y(m-1) + d_2(m)y(m-2) + d_3(m)y(m-3) + d_4(m)y(m-4) + Ge(m) \tag{16.53}$$

where the combined gain $G = G_1 G_2$. The poles of the fourth order predictor model of Equation (16.53) are the roots of the following polynomial:

$$\begin{aligned} &1 - d_1(m)z^{-1} - d_2(m)z^{-2} - d_3(m)z^{-3} - d_4(m)z^{-4} = \\ &= \left[1 - z^{-1}\alpha_1(m)\right]\left[1 - z^{-1}\alpha_2(m)\right]\left[1 - z^{-1}\beta_1\right]\left[1 - z^{-1}\beta_2\right] = 0 \end{aligned} \tag{16.54}$$

In Equation (16.54) 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 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.

16.4 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 $\hat{\boldsymbol{h}}$ is defined as

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{h}}|\boldsymbol{y}) &= \int\limits_{\boldsymbol{H}}\int\limits_{\boldsymbol{X}} C(\hat{\boldsymbol{h}},\boldsymbol{h})f_{\boldsymbol{X},\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{h}\,|\boldsymbol{y}\,)d\boldsymbol{x}\;d\boldsymbol{h}\\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})}\int\limits_{\boldsymbol{H}} C(\hat{\boldsymbol{h}},\boldsymbol{h})f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y}\,|\boldsymbol{h}\,)f_{\boldsymbol{H}}(\boldsymbol{h})d\boldsymbol{h}\end{aligned} \tag{16.55}$$

where $C(\hat{\boldsymbol{h}},\boldsymbol{h})$ is the cost of estimating the channel $\boldsymbol{h}$ as $\hat{\boldsymbol{h}}$, $f_{\boldsymbol{X},\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{h}|\boldsymbol{y})$ is the joint posterior density of the channel $\boldsymbol{h}$ and the channel input $\boldsymbol{x}$, $f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y}|\boldsymbol{h})$ is the observation likelihood, and $f_{\boldsymbol{H}}(\boldsymbol{h})$ is the prior pdf of the channel. The Bayesian estimate is obtained by minimisation of the risk function $\mathcal{R}(\hat{\boldsymbol{h}}|\boldsymbol{y})$. 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

$$\boldsymbol{y}(m) = \boldsymbol{x}(m) + \boldsymbol{h} \tag{16.56}$$

where the cepstral vectors $\boldsymbol{x}(m)$, $\boldsymbol{y}(m)$ and $\boldsymbol{h}$ are the channel input, the channel output and the channel respectively.

16.4.1 Conditional Mean Channel Estimation

A commonly used cost function in the Bayesian risk of Equation (16.55) is the mean square error $C(\boldsymbol{h}-\hat{\boldsymbol{h}}) = |\boldsymbol{h}-\hat{\boldsymbol{h}}|^2$, which results in the conditional mean (CM) estimate defined as

$$\hat{\boldsymbol{h}}^{CM} = \int\limits_{\boldsymbol{H}} \boldsymbol{h} f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{y})d\boldsymbol{h} \tag{16.57}$$

The posterior density of the channel input signal may be conditioned on an estimate of the channel vector $\hat{\boldsymbol{h}}$ and expressed as $f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{h}})$. The conditional mean of the channel input signal given the channel output $\boldsymbol{y}$ and an estimate of the channel $\hat{\boldsymbol{h}}$ is

$$\begin{aligned}\hat{\boldsymbol{x}}^{CM} &= \mathcal{E}[\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{h}}]\\ &= \int\limits_{\boldsymbol{X}} \boldsymbol{x} f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x}|\boldsymbol{y},\hat{\boldsymbol{h}})dx\end{aligned} \tag{16.58}$$

Equations (16.57) and (16.58) suggest a two-stage iterative method for channel estimation and the recovery of the channel input signal.

16.4.2 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 $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, the likelihood of a sequence of N P-dimensional channel output vectors $\{\boldsymbol{y}(m)\}$ given a channel input vector $\boldsymbol{h}$ is

$$\begin{aligned}f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)\,|\boldsymbol{h}\,) &= \prod_{m=0}^{N-1} f_{\boldsymbol{X}}(\boldsymbol{y}(m)-\boldsymbol{h})\\ &= \prod_{m=0}^{N-1}\frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{xx}|^{1/2}}\exp\left\{-0.5\,[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]^{\mathrm{T}}\boldsymbol{\Sigma}_{xx}^{-1}[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]\right\}\end{aligned} \tag{16.59}$$

To obtain the ML estimate of the channel $\boldsymbol{h}$, the derivative of the log likelihood function $\ln f_Y(\boldsymbol{y}|\boldsymbol{h})$ with respect to $\boldsymbol{h}$ is set to zero to yield

$$\hat{\boldsymbol{h}}^{ML} = \frac{1}{N}\sum_{m=0}^{N-1}(\boldsymbol{y}(m) - \boldsymbol{\mu}_x) \tag{16.60}$$

16.4.3 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 $\boldsymbol{Y} = \{\boldsymbol{y}(0), \ldots, \boldsymbol{y}(N-1)\}$ is

$$\begin{aligned} f_{H|Y}\big(\boldsymbol{h}\,|\boldsymbol{y}(0), \ldots, \boldsymbol{y}(N-1)\big) &= \prod_{m=0}^{N-1}\frac{1}{f_Y(\boldsymbol{y}(m))}f_{Y|H}\big(\boldsymbol{y}(m)\,|\boldsymbol{h}\big)f_H(\boldsymbol{h}) \\ &= \prod_{m=0}^{N-1}\frac{1}{f_Y(\boldsymbol{y}(m))}f_X\big(\boldsymbol{y}(m) - \boldsymbol{h}\big)f_H(\boldsymbol{h}) \end{aligned} \tag{16.61}$$

Assuming that the channel input $\boldsymbol{x}(m)$ is Gaussian, $f_X(\boldsymbol{x}(m)) = \mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$, with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the channel $\boldsymbol{h}$ is also Gaussian, $f_H(\boldsymbol{h}) = \mathcal{N}(\boldsymbol{h}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_{hh})$, with mean vector $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Sigma}_{hh}$, the logarithm of the posterior pdf is

$$\begin{aligned} \ln f_{H|Y}\left(\boldsymbol{h}\,|\boldsymbol{y}(0), \ldots, \boldsymbol{y}(N-1)\right) &= -\sum_{m=0}^{N-1}\ln f\left(\boldsymbol{y}(m)\right) - NP\ln(2\pi) - \frac{1}{2}\ln\left(|\boldsymbol{\Sigma}_{xx}|\,|\boldsymbol{\Sigma}_{hh}|\right) \\ -\sum_{m=0}^{N-1}\frac{1}{2}\Big\{[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_x]^{\mathrm{T}}\boldsymbol{\Sigma}_{xx}^{-1}[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_x] &+ (\boldsymbol{h} - \boldsymbol{\mu}_h)^{\mathrm{T}}\boldsymbol{\Sigma}_{hh}^{-1}(\boldsymbol{h} - \boldsymbol{\mu}_h)\Big\} \end{aligned} \tag{16.62}$$

The MAP channel estimate, obtained by setting the derivative of the log posterior function $\ln f_{H|Y}(\boldsymbol{h}|\boldsymbol{y})$ to zero, is

$$\hat{\boldsymbol{h}}^{MAP} = (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1}\boldsymbol{\Sigma}_{hh}(\bar{\boldsymbol{y}} - \boldsymbol{\mu}_x) + (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1}\boldsymbol{\Sigma}_{xx}\boldsymbol{\mu}_h \tag{16.63}$$

where

$$\bar{\boldsymbol{y}} = \frac{1}{N}\sum_{m=0}^{N-1}\boldsymbol{y}(m) \tag{16.64}$$

is the time-averaged estimate of the mean of observation vector. Note that for a Gaussian process the MAP and conditional mean estimates are identical.

16.4.4 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 16.8, is in recognition of speech distorted by a communication channel or a microphone. A hidden Markov model (HMM) 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 non-stationary process, such as speech, as a chain of N stationary states connected by a set of Markovian state transitions. The likelihood

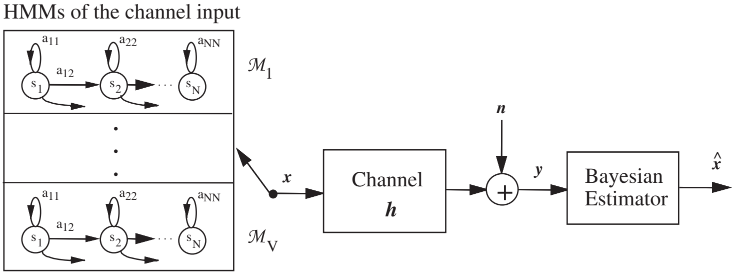


Figure 16.8 Illustration of a channel with the input modelled by a set of HMMs.

of an HMM $\mathcal{M}_i$ and a sequence of N P-dimensional channel input vectors $\boldsymbol{X}=[\boldsymbol{x}(0),\ldots,\boldsymbol{x}(N-1)]$ can be expressed in terms of the state transition and the observation pdfs of $\mathcal{M}_i$ as

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_i)=\sum_{\mathbf{s}} f_{\boldsymbol{X}|\mathcal{M},\boldsymbol{S}}(\boldsymbol{X}|\mathcal{M}_i,\mathbf{s})P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}_i) \tag{16.65}$$

where $f_{\boldsymbol{X}|\mathcal{M},\boldsymbol{S}}(\boldsymbol{X}|\mathcal{M}_i,\mathrm{s})$ is the likelihood that the sequence $\boldsymbol{X}=[\boldsymbol{x}(0),\ldots,\boldsymbol{x}(N-1)]$ was generated by the state sequence $\boldsymbol{s}=[s(0),\ldots,s(N-1)]$ of the model $\mathcal{M}_i$, and $P_{s|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}_i)$ is the Markovian prior pmf of the state sequence $\boldsymbol{s}$. 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

$$a_{ij}=P\big(s(m)=j|s(m-1)=i\big) \tag{16.66}$$

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

$$f_{\boldsymbol{X}|\mathcal{M},\mathrm{S}}(\boldsymbol{x}|\mathcal{M}_i,s)=\frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx},s}|^{1/2}}\exp\left\{-\frac{1}{2}[\boldsymbol{x}-\boldsymbol{\mu}_{x,s}]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx},s}^{-1}[\boldsymbol{x}-\boldsymbol{\mu}_{x,s}]\right\} \tag{16.67}$$

where $\boldsymbol{\mu}_{\boldsymbol{x},s}$ and $\boldsymbol{\Sigma}_{\boldsymbol{xx},s}$ 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 equalisation problem can be stated as follows: Given a sequence of N P-dimensional channel output vectors $\boldsymbol{Y}=[\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)]$, and the prior knowledge that the channel input sequence is drawn from a set of V HMMs $\mathcal{M}=\{\mathcal{M}_i\ i=1,\ldots,\ V\}$, estimate the channel response and the channel input.

The joint posterior pdf of an input word $\mathcal{M}_i$ and the channel vector $\boldsymbol{h}$ can be expressed as

$$f_{\mathcal{M},\boldsymbol{H}|\boldsymbol{Y}}(\mathcal{M}_i,\boldsymbol{h}|\boldsymbol{Y})=P_{\mathcal{M}|\boldsymbol{H},\boldsymbol{Y}}(\mathcal{M}_i\,|\boldsymbol{h},\boldsymbol{Y}\,)f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{Y}) \tag{16.68}$$

Simultaneous joint estimation of the channel vector $\boldsymbol{h}$ and classification of the unknown input word $\mathcal{M}_i$ is a non-trivial 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' rule, the posterior pdf of the channel $\boldsymbol{h}$ conditioned on the assumption that the input model is $\mathcal{M}_i$ and given the observation sequence $\boldsymbol{Y}$ can be expressed as

$$f_{\boldsymbol{H}|\mathcal{M},\boldsymbol{Y}}(\boldsymbol{h}\,|\mathcal{M}_i,\boldsymbol{Y}\,)=\frac{1}{f_{\boldsymbol{Y}|\mathcal{M}}(\boldsymbol{Y}|\mathcal{M}_i)}f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y}\,|\mathcal{M}_i,\boldsymbol{h}\,)f_{\boldsymbol{H}|\mathcal{M}}(\boldsymbol{h}|\mathcal{M}_i) \tag{16.69}$$

The likelihood of the observation sequence, given the channel and the input word model, can be expressed as

$$f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y}|\mathcal{M}_i,\boldsymbol{h})=f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{Y}-\boldsymbol{h}|\mathcal{M}_i) \tag{16.70}$$

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 $\boldsymbol{s} = [s(0),\ s(1),\ \ldots,\ s(N-1)]$, the pdf of a sequence of N independent observation vectors $\boldsymbol{Y} = [\boldsymbol{y}(0),\ \boldsymbol{y}(1),\ \ldots,\ \boldsymbol{y}(N-1)]$ is

$$\begin{aligned} f_{Y|\boldsymbol{H},\mathbf{S},\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{h},\mathbf{s},\mathcal{M}_i) &= \prod_{m=0}^{N-1} f_{X|S,\mathbf{M}}\left(\boldsymbol{y}\left(m\right)-\boldsymbol{h}|s\left(m\right),\mathcal{M}_i\right) \\ &= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}|^{1/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]\right\} \end{aligned} \tag{16.71}$$

Taking the derivative of the log-likelihood of Equation (16.71) with respect to the channel vector $\boldsymbol{h}$ yields a maximum likelihood channel estimate as

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \sum_{m=0}^{N-1}\left(\sum_{k=0}^{N-1}\boldsymbol{\Sigma}_{xx,s(k)}^{-1}\right)^{-1}\boldsymbol{\Sigma}_{xx,s(m)}^{-1}\left(\boldsymbol{y}(m)-\boldsymbol{\mu}_{x,s(m)}\right) \tag{16.72}$$

Note that when all the state observation covariance matrices are identical the channel estimate becomes

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \frac{1}{N}\sum_{m=0}^{N-1}\left(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right) \tag{16.73}$$

The ML estimate of Equation (16.73) is based on the ML state sequence $\boldsymbol{s}$ of $\mathcal{M}_i$. In the following section we consider the conditional mean estimate over all state sequences of a model.

16.4.5 MAP Channel Estimate Based on HMMs

The conditional pdf of a channel $\boldsymbol{h}$ averaged over all HMMs can be expressed as

$$f_{\boldsymbol{H}|Y}(\boldsymbol{h}|\boldsymbol{Y}) = \sum_{i=1}^{V}\sum_{\boldsymbol{s}} f_{\boldsymbol{H}|\boldsymbol{Y},\boldsymbol{S},\mathcal{M}}\left(\boldsymbol{h}\,|\boldsymbol{Y},\boldsymbol{s},\mathcal{M}_i\right) P_{\boldsymbol{S}|\mathcal{M}}\left(\boldsymbol{s}\,|\mathcal{M}_i\right) P_{\mathcal{M}}(\mathcal{M}_i) \tag{16.74}$$

where $P_{\mathcal{M}}(\mathcal{M}_i)$ is the prior pmf of the input words. Given a sequence of N P-dimensional observation vectors $\boldsymbol{Y} = [\boldsymbol{y}(0),\ \ldots,\ \boldsymbol{y}(N-1)]$, the posterior pdf of the channel $\boldsymbol{h}$ along a state sequence $\boldsymbol{s}$ of an HMM $\mathcal{M}_i$ is defined as

$$\begin{aligned} f_{Y|\boldsymbol{H},\mathbf{S},\mathbf{M}}\left(\boldsymbol{h}\,|\boldsymbol{Y},\mathbf{s},\mathcal{M}_i\right) &= \frac{1}{f_Y\left(\boldsymbol{Y}\right)} f_{Y|\boldsymbol{H},\mathbf{S},\mathcal{M}}\left(\boldsymbol{Y}\,|\boldsymbol{h},\mathbf{s},\mathcal{M}_i\right) f_{\boldsymbol{H}}\left(\boldsymbol{h}\right) \\ &= \frac{1}{f_Y\left(\boldsymbol{Y}\right)} \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P}\left|\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}\right|^{1/2}\left|\boldsymbol{\Sigma}_{\boldsymbol{hh}}\right|^{1/2}} \\ &\quad \times \exp\left\{-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]\right\} \\ &\quad \times \exp\left[-\frac{1}{2}\left(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}}\right)^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\left(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}}\right)\right] \end{aligned} \tag{16.75}$$

where it is assumed that each state of the HMM has a Gaussian distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{x},s(m)}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}$, and that the channel $\boldsymbol{h}$ is also Gaussian-distributed, with mean vector $\boldsymbol{\mu}_{\boldsymbol{h}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{hh}}$. The MAP estimate along state $\boldsymbol{s}$, on the left-hand side of Equation (16.75), can be obtained as

$$\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y},\mathbf{s},\mathcal{M}_i)=\sum_{m=0}^{N-1}\left[\sum_{k=0}^{N-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{xx},s(k)}^{-1}+\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\right)\right]^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]$$
$$+\left[\sum_{k=0}^{N-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{xx},s(k)}^{-1}+\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\right)\right]^{-1}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\boldsymbol{\mu}_{\boldsymbol{h}} \tag{16.76}$$

The MAP estimate of the channel over all state sequences of all HMMs can be obtained as

$$\hat{\boldsymbol{h}}(\boldsymbol{Y})=\sum_{i=1}^{V}\sum_{\mathbf{S}}\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y},\mathbf{s},\mathcal{M}_i)P_{\mathbf{S}|\mathcal{M}}(\mathbf{s}|\mathcal{M}_i)P_{\mathcal{M}}(\mathcal{M}_i) \tag{16.77}$$

16.4.6 Implementations of HMM-Based Deconvolution

In this section, we consider three implementation methods for HMM-based channel equalisation.

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

$$\boldsymbol{\mu}_{\boldsymbol{x}}=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\boldsymbol{\mu}_{\mathcal{M}_i,j},\quad \boldsymbol{\Sigma}_{\boldsymbol{xx}}=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\boldsymbol{\Sigma}_{\mathcal{M}_i,j} \tag{16.78}$$

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, V and N_s denote the number of models and number of states per model respectively. The maximum likelihood estimate of the channel, $\hat{\boldsymbol{h}}^{ML}$, is defined as

$$\hat{\boldsymbol{h}}^{ML}=(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_{\mathbf{x}}) \tag{16.79}$$

where $\bar{\boldsymbol{y}}$ is the time-averaged channel output. The estimate of the channel input is

$$\hat{\boldsymbol{x}}(m)=\boldsymbol{y}(m)-\hat{\boldsymbol{h}}^{ML} \tag{16.80}$$

Using the averages over all states and models, the MAP channel estimate becomes

$$\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y})=\sum_{m=0}^{N-1}(\boldsymbol{\Sigma}_{\boldsymbol{xx}}+\boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{hh}}(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{x}})+(\boldsymbol{\Sigma}_{\boldsymbol{xx}}+\boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\,\boldsymbol{\mu}_{\boldsymbol{h}} \tag{16.81}$$

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 $\hat{\boldsymbol{h}}_w$ 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 16.9):

For $i=1$ to number of words V {

Step 1: Using each HMM, $\mathcal{M}_i$, make an estimate of the channel, $\hat{\boldsymbol{h}}_i$,
Step 2: Using the channel estimate, $\hat{\boldsymbol{h}}_i$, estimate the channel input $\hat{\boldsymbol{x}}(m)=\boldsymbol{y}(m)-\hat{\boldsymbol{h}}_i$
Step 3: Compute a probability score for model $\mathcal{M}_i$, given the estimate $[\hat{\mathbf{x}}(m)]$. }

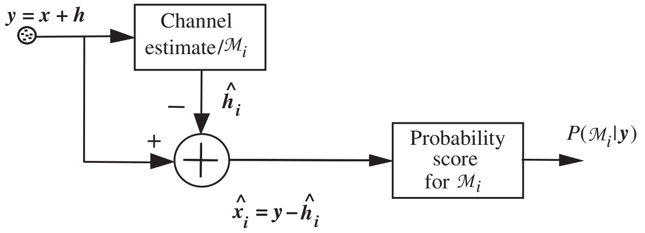

$y = x + h$
Channel estimate/$\mathcal{M}_i$
$-$
$\hat{h}_i$
$+$
$\hat{x}_i = y - \hat{h}_i$
Probability score for $\mathcal{M}_i$
$P(\mathcal{M}_i|y)$

Figure 16.9 Hypothesised channel estimation procedure.

Select the channel estimate associated with the most probable word.

Figure 16.10 shows the ML channel estimates of two channels using unweighted average and hypothesised-input methods.

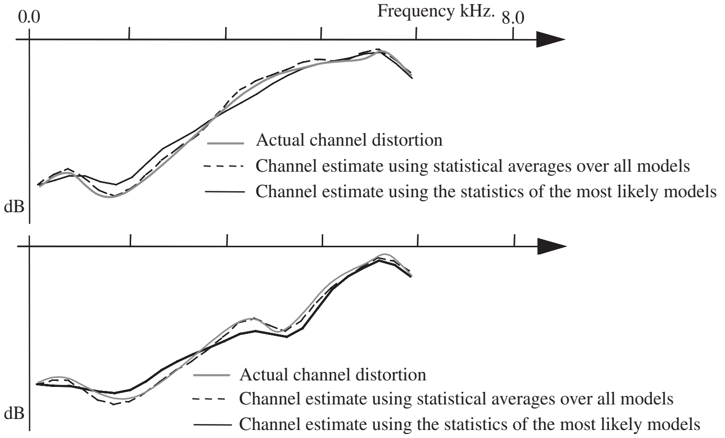


Figure 16.10 Illustration of actual and estimated channel response for two channels.

Method III: Decision-Directed Equalisation

Blind adaptive equalisers are often composed of two distinct sections: an adaptive linear equaliser followed by a non-linear estimator to improve the equaliser output. The output of the non-linear 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 non-linear estimator as the desired signal assumes that the linear equalisation filter removes a large part of the channel distortion, thereby enabling the non-linear 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 16.11 illustrates a blind equaliser incorporating an adaptive linear filter followed by a hidden Markov model classifier/estimator. The HMM classifies the output of the filter as one of a 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

$$z(m) = x(m) + v(m) \tag{16.82}$$

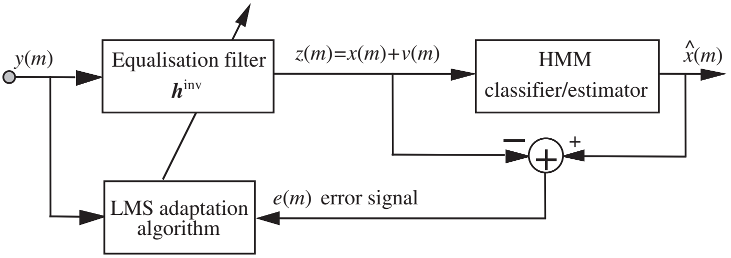


Figure 16.11 A decision-directed equaliser.

The HMM may incorporate state-based Wiener filters for suppression of the convolutional noise $v(m)$ as described in Section 5.6. Assuming that the LMS adaptation method is employed, the adaptation of the equaliser coefficient vector is governed by the following recursive equation:

$$\hat{h}^{\text{inv}}(m) = \hat{h}^{\text{inv}}(m-1) + \mu e(m)y(m) \tag{16.83}$$

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

$$e(m) = \hat{x}^{HMM}(m) - z(m) \tag{16.84}$$

where $\hat{x}^{HMM}$ (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.

16.5 Blind Equalisation for Digital Communication Channels

High speed transmission of digital data over analog channels, such as telephone lines or a radio channels, requires adaptive equalisation to reduce decoding errors caused by channel distortions. In telephone lines, the channel distortions are due to the non-ideal magnitude response and the nonlinear phase response of the lines. In radio channel environments, the distortions are due to non-ideal 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 16.12.

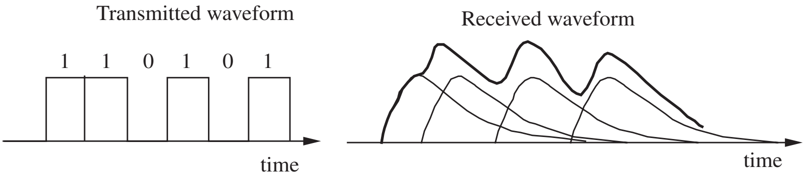


Figure 16.12 Illustration of inter-symbol interference in a binary pulse amplitude modulation system.

In a digital communication system, the transmitter modem takes N bits of binary data at a time, and encodes them into one of 2^N analog symbols for transmission, at the signalling rate, over an analog channel. At the receiver the analog 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 multi-level pulse amplitude modulation (M-ary PAM) as a convenient scheme for the study of adaptive channel equalisation.

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

$$x(t) = \sum_k a(k) r(t - kT_s) \tag{16.85}$$

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:

$$y(t) = \int_{-\infty}^{\infty} h(\tau) x(t - \tau) d\tau \tag{16.86}$$

where $h(t)$ is the channel impulse response. The sampled version of the channel output is given by the following discrete-time equation:

$$y(m) = \sum_k h_k \, x(m - k) \tag{16.87}$$

To remove the channel distortion, the sampled channel output $y(m)$ is passed to an equaliser with an impulse response $\hat{h}_k^{\text{inv}}$. The equaliser output $z(m)$ is given as

$$\begin{aligned} z(m) &= \sum_k \hat{h}_k^{\text{inv}} y(m - k) \\ &= \sum_j x(m - j) \sum_k \hat{h}_k^{\text{inv}} h_{j-k} \end{aligned} \tag{16.88}$$

where Equation (16.87) is used to obtain the second line of Equation (16.88). 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 (16.88), the channel distortion would be cancelled if

$$h_m^c = h_m * \hat{h}_m^{\text{inv}} = \delta(m - D) \tag{16.89}$$

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

$$h^c(kT_s + D) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \tag{16.90}$$

Note that in Equation (16.90), at the sampling instants the channel distortion is cancelled, and hence there is no ISI at the sampling instants. A function that satisfies Equation (16.90) 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 (16.90), and also to jitters in the synchronisation and the sampling process.

16.5.1 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

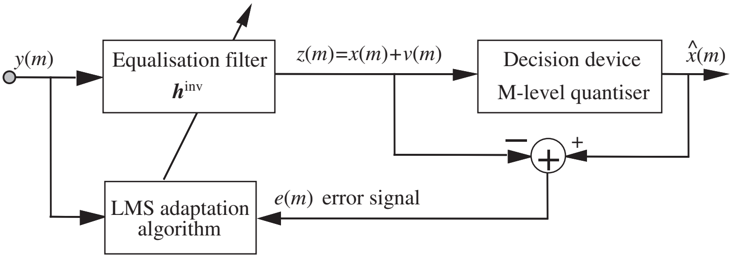


Figure 16.13 Configuration of an adaptive channel equaliser with an estimate of the channel input used as an 'internally' generated desired signal.

equalisation, requires an internally generated desired signal as illustrated in Figure 16.13. Digital blind equalisers are composed of two distinct sections: an adaptive equaliser that removes a large part of the channel distortion, followed by a non-linear estimator for an improved estimate of the channel input. The output of the non-linear estimator is the final estimate of the channel input, and is used as the desired signal *to direct* the equaliser adaptation. A method of ensuring that the equaliser 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:

$$\hat{\boldsymbol{h}}^{\mathrm{inv}}(m) = \hat{\boldsymbol{h}}^{\mathrm{inv}}(m-1) + \mu e(m)\boldsymbol{y}(m) \tag{16.91}$$

where $\hat{\boldsymbol{h}}^{\mathrm{inv}}(m)$ is an estimate of the optimal inverse channel filter $\boldsymbol{h}^{\mathrm{inv}}$, the scalar μ is the adaptation step size, and the error signal $e(m)$ is defined as

$$\begin{aligned} e(m) &= \psi\,(z(m)) - z(m) \\ &= \hat{x}(m) - z(m) \end{aligned} \tag{16.92}$$

where $\hat{x}(m) = \psi\,(z(m))$ is a non-linear estimate of the channel input. For example, in a binary communication system with an input alphabet $\{\pm a\}$ we can use a signum non-linearity such that $\hat{x}(m) = a.sgn\,(z(m))$ where the function $sgn(\cdot)$ gives the sign of the argument. In the following, we use a Bayesian framework to formulate the nonlinear estimator $\psi()$.

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:

$$z(m) = x(m) + v(m) \tag{16.93}$$

where $v(m)$ is the so-called convolutional noise defined as

$$\begin{aligned} v(m) &= x(m) - \sum_k \hat{h}_k^{\mathrm{inv}} y(m-k) \\ &= \sum_k (h_k^{\mathrm{inv}} - \hat{h}_k^{\mathrm{inv}}) y(m-k) \end{aligned} \tag{16.94}$$

In the following, we assume that the non-linear 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 equaliser, 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

$$f_E\,(e(m)) = \frac{1}{\sqrt{2\pi}\,\sigma_e} \exp\left(-\frac{e^2(m)}{2\sigma_e^2}\right) \tag{16.95}$$

where σ_e^2, the noise variance, can be estimated using the recursive time-update equation

$$\sigma_e^2(m) = \rho\sigma_e^2(m-1) + (1-\rho)e^2(m) \tag{16.96}$$

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

$$\hat{x}(m) = \arg\min_{\hat{x}(m)} \int_X C\left(x(m), \hat{x}(m)\right) f_{X|Z}\,(x(m)|z(m))\,dx(m) \tag{16.97}$$

where $C\left(x(m), \hat{x}(m)\right)$ 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 (16.97) yields the maximum a posteriori (MAP) estimate

$$\begin{aligned}\hat{x}^{MAP}(m) &= \arg\max_{x(m)} f_{X|Z}\left(x(m)|z(m)\right)\\ &= \arg\max_{x(m)} f_E\left(z(m) - x(m)\right) P_X\left(x(m)\right)\end{aligned} \tag{16.98}$$

Now, as an example consider an M-ary pulse amplitude modulation system, and let $\{a_i,\ i = 1, \ldots,\ M\}$ denote the set of M pulse amplitudes with a probability mass function

$$P_X\left(x(m)\right) = \sum_{i=1}^{M} P_i \delta\,(x(m) - a_i) \tag{16.99}$$

The pdf of the equaliser output $z(m)$ can be expressed as the mixture pdf

$$f_Z\,(z(m)) = \sum_{i=1}^{M} P_i f_E\,(x(m) - a_i) \tag{16.100}$$

The posterior density of the channel input is

$$P_{X|Z}\left(x(m) = a_i\,|z(m)\right) = \frac{1}{f_Z\,(z(m))} f_E\,(z(m) - a_i)\,P_X\,(x(m) = a_i) \tag{16.101}$$

and the MAP estimate is obtained from

$$\hat{x}^{MAP}(m) = \arg\max_{a_i}\,(f_E\,(z(m) - a_i)\,P_X\,(x(m) = a_i)) \tag{16.102}$$

Note that the classification of the continuous-valued equaliser output $z(m)$ into one of M discrete channel input symbols is basically a non-linear process. Substitution of the zero-mean Gaussian model for the convolutional noise $e(m)$ in Equation (16.102) yields

$$\hat{x}^{MAP}(m) = \arg\max_{a_i}\left[P_X\,(x(m) = a_i)\exp\left\{-\frac{[z(m) - a_i]^2}{2\sigma_e^2}\right\}\right] \tag{16.103}$$

Note that when the symbols are equi-probable, the MAP estimate reduces to a simple threshold decision device. Figure 16.13 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.

16.5.2 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

$$P\left(x(m)\right) = P_0 \delta\left(x(m) - a_0\right) + P_1 \delta\left(x(m) - a_1\right) \tag{16.104}$$

Assume that at the output of the linear adaptive equaliser in Figure 16.13, the convolutional noise $v(m)$ is a zero-mean Gaussian process with variance σ_v^2. Therefore the pdf of the equaliser output $z(m) = x(m) + v(m)$ is a mixture of two Gaussian pdfs and can be described as

$$f_Z\left(z(m)\right) = \frac{P_0}{\sqrt{2\pi}\sigma_v} \exp\left\{-\frac{[z(m)-a_0]^2}{2\sigma_v^2}\right\} + \frac{P_1}{\sqrt{2\pi}\sigma_v} \exp\left\{-\frac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \tag{16.105}$$

The MAP estimate of the channel input signal is

$$\hat{x}(m) = \begin{cases} a_0 & \text{if } \dfrac{P_0}{\sqrt{2\pi}\sigma_v} \exp\left\{-\dfrac{[z(m)-a_0]^2}{2\sigma_v^2}\right\} > \dfrac{P_1}{\sqrt{2\pi}\sigma_v} \exp\left\{-\dfrac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \\ a_1 & \text{otherwise} \end{cases} \tag{16.106}$$

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 $sgn(x(m))$, and the error signal is given by

$$e(m) = z(m) - sgn\left(z(m)\right) a \tag{16.107}$$

Figure 16.14 shows the error signal as a function of $z(m)$. An undesirable property of a hard non-linearity, such as the $sgn(\cdot)$ 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.

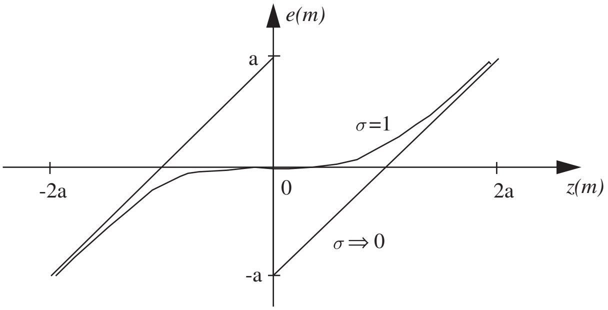


Figure 16.14 Comparison of the error functions produced by the hard non-linearity of a sign function Equation (17.107) and the soft non-linearity of Equation (16.108).

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 (1993) proposed a soft non-linearity of the following form

$$e(m) = z(m) - \frac{e^{2az(m)/\sigma^2} - 1}{e^{2az(m)/\sigma^2} + 1} a \tag{16.108}$$

The error $e(m)$ is small when the magnitude of $z(m)$ is small and large when magnitude of $z(m)$ is large.

16.6 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 always include the correct phase characteristics of the process. That is because the covariance matrix of a minimum phase and its equivalent maximum phase system are identical (for example try an experiment in Matlab with the covariance matrix of the outputs of $H_1(z) = G_1(1 - \alpha z^{-1})$ and $H_2(z) = G_2(1 - (1/\alpha)z^{-1})$ where $G_1 = 1/(1+\alpha)$ and $G_2 = 1/(1 + (1/\alpha))$ for unity gain. Therefore, given the channel output, it is not always 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 the higher-order statistics is their ability to model the phase characteristics. Further motivations are the potential of the higher order statistics to model channel non-linearities, and to estimate a non-Gaussian signal in a high level of Gaussian noise.

16.6.1 Higher-Order Moments, Cumulants and Spectra

The k^{th} *order* moment of a random variable X is defined as

$$\begin{aligned} m_k &= \mathcal{E}[x^k] \\ &= (-j)^k \left.\frac{\partial^k \Phi_X(\omega)}{\partial \omega^k}\right|_{\omega=0} \end{aligned} \tag{16.109}$$

where $\Phi_X(\omega)$ is the *characteristic function* of the random variable X defined as

$$\Phi_X(\omega) = \mathcal{E}[\exp(j\omega x)] \tag{16.110}$$

From Equations (16.109) and (16.110), 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

$$\mathcal{E}[x_1^{k_1} x_2^{k_2}] = (-j)^{k_1+k_2} \left.\frac{\partial^{k_1}\partial^{k_2}\Phi_{X_1X_2}(\omega_1,\omega_2)}{\partial \omega_1^{k_1}\partial \omega_2^{k_2}}\right|_{\omega_1=\omega_2=0} \tag{16.111}$$

and in general the joint k^{th} order moment of N random variables is defined as

$$\begin{aligned} m_k &= \mathcal{E}[x_1^{k_1} x_2^{k_2} \ldots {x_N}^{k_N}] \\ &= (-\mathrm{j})^k \left.\frac{\partial^k \Phi(\omega_1,\omega_2,\ldots,\omega_N)}{\partial \omega_1^{k_1}\partial \omega_2^{k_2}\cdots\partial \omega_N^{k_N}}\right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \end{aligned} \tag{16.112}$$

where $k = k_1 + k_2 + \ldots + k_N$ and the joint characteristic function is

$$\Phi(\omega_1,\omega_2,\ldots,\omega_N) = \mathcal{E}\left[\exp\left(j\omega_1 x_1 + j\omega_2 x_2 + \cdots + j\omega_N x_N\right)\right] \tag{16.113}$$

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

$$m_x(\tau_1, \tau_2, \ldots, \tau_{K-1}) = \mathcal{E}[x(m), x(m+\tau_1)x(m+\tau_2)\cdots x(m+\tau_{k-1})] \tag{16.114}$$

Note that the second-order moment $\mathcal{E}[x(m)x(m+\tau)]$ is the autocorrelation function.

16.6.1.1 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

$$C_X(\omega) = \ln \Phi_X(\omega) = \ln \mathcal{E}[\exp(j\omega x)] \tag{16.115}$$

Using a Taylor series expansion of the term $\mathcal{E}$ $[\exp(j\omega x)]$ in Equation (16.115) the cumulant generating function can be expanded as

$$C_X(\omega) = \ln\left(1 + m_1(\mathrm{j}\omega) + \frac{m_2}{2!}(\mathrm{j}\omega)^2 + \frac{m_3}{3!}(\mathrm{j}\omega)^3 + \cdots + \frac{m_n}{n!}(\mathrm{j}\omega)^n + \cdots\right) \tag{16.116}$$

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

$$c_k = (-\mathrm{j})^k \left.\frac{\partial^k C_X(\omega)}{\partial \omega^k}\right|_{\omega=0} \tag{16.117}$$

From Equations (16.116) and (16.117), we have

$$c_1 = m_1 \tag{16.118}$$

$$c_2 = m_2 - m_1^2 \tag{16.119}$$

$$c_3 = m_3 - 3m_1 m_2 + 2m_1^2 \tag{16.120}$$

and so on. The general form of the k^{th} order $(k = k_1 + k_2 + \cdots + k_N)$ joint cumulant generating function is

$$c_{k_1 \cdots k_N} = (-\mathrm{j})^{k_1+\cdots+k_N} \left.\frac{\partial^{k_1+\cdots+k_N} \ln \Phi_X(\omega_1, \cdots, \omega_N)}{\partial \omega_1^{k_1} \cdots \partial \omega_N^{k_N}}\right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \tag{16.121}$$

The cumulants of a zero mean random process $x(m)$ are given as

$$c_x = \mathcal{E}[x(k)] = m_x = 0 \qquad \text{(mean)} \tag{16.122}$$

$$\begin{aligned} c_x(k) &= \mathcal{E}[x(m)x(m+k)] - \mathcal{E}[x(m)]^2 \\ &= m_x(k) - m_x^2 = m_x(k) \end{aligned} \qquad \text{(covariance)} \tag{16.123}$$

$$\begin{aligned} c_x(k_1, k_2) &= m_x(k_1, k_2) - m_x[m_x(k_1) + m_x(k_2) + m_x(k_2 - k_1)] + 2\,(m_x)^3 \\ &= m_x(k_1, k_2) \end{aligned} \qquad \text{(skewness)} \tag{16.124}$$

$$\begin{aligned} c_x(k_1, k_2, k_3) &= m_x(k_1, k_2, k_3) - m_x(k_1)m_x(k_3 - k_2) \\ &\quad - m_x(k_2)m_x(k_3 - k_1) - m_x(k_3)m_x(k_2 - k_1) \end{aligned} \tag{16.125}$$

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 (1985)) is defined as

$$c_x(k_1,k_2,\dots,k_n) = m_x(k_1,k_2,\dots,k_n) - m_x^G(k_1,k_2,\dots,k_n)\ for\ n = 3,4,... \tag{16.126}$$

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 (16.126), it follows that for a Gaussian process, the cumulants of order greater than 2 are identically zero.

16.6.1.2 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

$$C_X(\omega_1,\dots,\omega_{k-1}) = \frac{1}{(2\pi)^{k-1}} \sum_{\tau_1=-\infty}^{\infty} \cdots \sum_{\tau_{k-1}=-\infty}^{\infty} c_x(\tau_1,\dots,\tau_{k-1}) e^{-j(\omega_1\tau_1+\cdots+\omega_{k-1}\tau_{k-1})} \tag{16.127}$$

For the case $k=2$, the second-order spectrum is the power spectrum given as

$$C_X(\omega) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{\infty} c_x(\tau) e^{-j\omega\tau} \tag{16.128}$$

The *bi-spectrum* is defined as

$$C_X(\omega_1,\omega_2) = \frac{1}{(2\pi)^2} \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} c_x(\tau_1,\tau_2) e^{-j(\omega_1\tau_1+\omega_2\tau_2)} \tag{16.129}$$

and the *tri-spectrum* is

$$C_X(\omega_1,\omega_2,\omega_3) = \frac{1}{(2\pi)^3} \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} \sum_{\tau_3=-\infty}^{\infty} c_x(\tau_1,\tau_2,\tau_3) e^{-j(\omega_1\tau_1+\omega_2\tau_2+\omega_3\tau_3)} \tag{16.130}$$

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.

16.6.2 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

$$C_Y(\omega_1,\dots,\omega_{k-1}) = H(\omega_1)\cdots H(\omega_{k-1})H^*(\omega_1+\cdots+\omega_{k-1})C_X(\omega_1,\dots,\omega_{k-1}) \tag{16.131}$$

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

$$|C_Y(\omega_1,\dots,\omega_{k-1})| = |H(\omega_1)|\cdots|H(\omega_{k-1})|\,|H(\omega_1+\cdots+\omega_{k-1})|\,|C_X(\omega_1,\dots,\omega_{k-1})| \tag{16.132}$$

and the phase of the k^{th}-order spectrum is

$$\Phi_Y(\omega_1,\dots,\omega_{k-1}) = \Phi_H(\omega_1)+\cdots+\Phi_H(\omega_{k-1}) - \Phi_H(\omega_1+\cdots+\omega_{k-1}) + \Phi_X(\omega_1,\dots,\omega_{k-1}) \tag{16.133}$$

16.6.3 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

$$H(z) = GH_{\min}(z)H_{\max}(z^{-1})z^{-D} \tag{16.134}$$

$$H_{\min}(z) = \prod_{i=1}^{P_1}(1-\alpha_i z^{-1}),\ |\alpha_i| < 1 \tag{16.135}$$

$$H_{\max}(z^{-1}) = \prod_{i=1}^{P_2}(1-\beta_i z),\ |\beta_i| < 1 \tag{16.136}$$

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 (16.134) causal. The complex cepstrum of $H(z)$ is defined as

$$h_c(m) = Z^{-1}\left(\ln H(z)\right) \tag{16.137}$$

where Z^{-1} denotes the inverse z-transform. At $z = e^{j\omega}$, the z-transform is the discrete Fourier transform (DFT), 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

$$P_Y(\omega) = \frac{\sigma_x^2}{2\pi}H(\omega)H^*(\omega) \tag{16.138}$$

The cepstrum of the power spectrum of $y(m)$ is defined as

$$\begin{aligned} y_c(m) &= IDFT\left(\ln P_Y(\omega)\right) \\ &= IDFT\left(\ln\left(\sigma_x^2 G^2/2\pi\right) + \ln H_{\min}(\omega) + H_{\max}(-\omega) + \ln H^*_{\min}(\omega) + H^*_{\max}(-\omega)\right) \end{aligned} \tag{16.139}$$

where IDFT is the inverse discrete Fourier transform. Substituting Equations (16.135) and (16.36) in (16.139), the cepstrum can be expressed as

$$y_c(m) = \begin{cases} \ln\left(G^2\sigma_x^2/2\pi\right), & m=0 \\ -\left(A^{(m)}+B^{(m)}\right)/m, & m>0 \\ \left(A^{(-m)}+B^{(-m)}\right)/m, & m<0 \end{cases} \tag{16.140}$$

where $A^{(m)}$ and $B^{(m)}$ are defined as

$$A^{(m)} = \sum_{i=1}^{P_1}\alpha_i^m \tag{16.141}$$

$$B^{(m)} = \sum_{i=1}^{P_2}\beta_i^m \tag{16.142}$$

Note from Equation (16.140) that, along the index m, the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ overlap and cannot be separated.

16.6.3.1 Bi-Cepstrum

The bi-cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the bi-spectrum:

$$y_c(m_1, m_2) = IDFT_2[\log C_Y(\omega_1, \omega_2)] \tag{16.143}$$

where $IDFT_2[.]$ denotes the two-dimensional inverse discrete Fourier transform. The relationship between the bi-spectra of the input and output of a linear system is

$$C_Y(\omega_1, \omega_2) = H(\omega_1)H(\omega_2)H^*(\omega_1 + \omega_2)C_X(\omega_1, \omega_2) \tag{16.144}$$

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

$$\begin{aligned} C_Y(\omega_1, \omega_2) = &\frac{\gamma_x^{(3)}G^3}{(2\pi)^2} H_{\min}(\omega_1)H_{\max}(-\omega_1)H_{\min}(\omega_2)H_{\max}(-\omega_2) \\ &\times H^*_{\min}(\omega_1 + \omega_2)H^*_{\max}(-\omega_1 - \omega_2) \end{aligned} \tag{16.145}$$

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 (16.145) yields

$$\begin{aligned} \ln C_y(\omega_1, \omega_2) = &\ln|A| + \ln H_{\min}(\omega_1) + \ln H_{\max}(-\omega_1) + \ln H_{\min}(\omega_2) + \ln H_{\max}(-\omega_2) \\ &+ \ln H^*_{\min}(\omega_1 + \omega_2) + \ln H^*_{\max}(-\omega_1 - \omega_2) \end{aligned} \tag{16.146}$$

where $A = \gamma_x^{(3)}G^3/(2\pi)^2$. The bi-cepstrum is obtained through the inverse Discrete Fourier transform of Equation (16.146) as

$$y_c(m_1, m_2) = \begin{cases} \ln|A|, & m_1 = m_2 = 0 \\ -A^{(m_1)}\big/m_1, & m_1 > 0, m_2 = 0 \\ -A^{(m_2)}\big/m_2, & m_2 > 0, m_1 = 0 \\ -B^{(-m_1)}\big/m_1, & m_1 < 0, m_2 = 0 \\ B^{(-m_2)}\big/m_2, & m_2 < 0, m_1 = 0 \\ -B^{(m_2)}\big/m_2, & m_1 = m_2 > 0 \\ A^{(-m_2)}\big/m_2, & m_1 = m_2 < 0 \\ 0, & \text{otherwise} \end{cases} \tag{16.147}$$

Note from Equation (16.147) 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 (16.147) 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

$$y_c(m_1, m_2) * \left[-m_1 c_y(m_1, m_2)\right] = -m_1 c_y(m_1, m_2) \tag{16.148}$$

Substituting Equation (16.147) in Equation (16.148) yields

$$\begin{aligned} &\sum_{i=1}^{\infty} A^{(i)}[c_x(m_1 - i, m_2) - c_x(m_1 + i, m_2 + i)] + B^{(i)} \\ &[c_x(m_1 - i, m_2 - i) - c_x(m_1 + i, m_2)] = -m_1 c_x(m_1, m_2) \end{aligned} \tag{16.149}$$

The truncation of the infinite summation in Equation (16.149) provides an approximate equation as

$$\begin{aligned}&\sum_{i=1}^{P} A^{(i)}[c_x(m_1-i,m_2)-c_x(m_1+i,m_2+i)]+\\&\sum_{i=1}^{Q} B^{(i)}[c_x(m_1-i,m_2-i)-c_x(m_1+i,m_2)]\approx -m_1c_x(m_1,m_2)\end{aligned}\tag{16.150}$$

Equation (16.150) can be used to solve for the cepstral parameters $A^{(m)}$ and $B^{(m)}$.

16.6.3.2 Tri-Cepstrum

The tri-cepstrum of a signal $y(m)$ is defined as the inverse Fourier transform of the tri-spectrum:

$$y_c(m_1,m_2,m_3)=IDFT_3[\ln C_Y(\omega_1,\omega_2,\omega_3)]\tag{16.151}$$

where $IDFT_3[\cdot]$ denotes the three-dimensional inverse discrete Fourier transform. The tri-spectra of the input and output of the linear system are related by

$$C_Y(\omega_1,\omega_2,\omega_3)=H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3)C_X(\omega_1,\omega_2,\omega_3)\tag{16.152}$$

Assuming that the channel input $x(m)$ is uncorrelated, Equation (16.152) becomes

$$C_Y(\omega_1,\omega_2,\omega_3)=\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3)\tag{16.153}$$

where $\gamma_x^{(4)}/(2\pi)^3$ is the fourth-order cumulant of the input signal. Taking the logarithm of the tri-spectrum gives

$$\begin{aligned}\ln C_Y(\omega_1,\omega_2,\omega_3)=&\ln(A)+\ln H_{\min}(\omega_1)+\ln H_{\max}(-\omega_1)+\ln H_{\min}(\omega_2)+\ln H_{\max}(-\omega_2)\\&+\ln H_{\min}(\omega_3)+\ln H_{\max}(-\omega_3)+\ln H^*_{\min}(\omega_1+\omega_2+\omega_3)\\&+\ln H^*_{\min}(-\omega_1-\omega_2-\omega_3)\end{aligned}\tag{16.154}$$

where $A=\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}$. From Equations (16.151) and (16.154), we have

$$y_c(m_1,m_2,m_3)=\begin{cases}\ln A, & m_1=m_2=m_3=0\\ -A^{(m_1)}\big/m_1, & m_1>0, m_2=m_3=0\\ -A^{(m_2)}\big/m_2, & m_2>0, m_1=m_3=0\\ -A^{(m_3)}\big/m_3, & m_3>0, m_1=m_2=0\\ B^{(-m_1)}\big/m_1, & m_1<0, m_2=m_3=0\\ B^{(-m_2)}\big/m_2, & m_2<0, m_1=m_3=0\\ B^{(-m_3)}\big/m_3, & m_3<0, m_1=m_2=0\\ -B^{(m_2)}\big/m_2, & m_1=m_2=m_3>0\\ A^{(m_2)}\big/m_2, & m_1=m_2=m_3<0\\ 0 & \text{otherwise}\end{cases}\tag{16.155}$$

Note from Equation (16.155) 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.

16.6.3.3 Calculation of Equaliser Coefficients from the Tri-cepstrum

Assuming that the channel z-transfer function can be described by Equation (16.134), the inverse channel can be written as

$$H^{\text{inv}}(z) = \frac{1}{H(z)} = \frac{1}{H_{\min}(z)H_{\max}(z^{-1})} = H^{\text{inv}}_{\min}(z)H^{\text{inv}}_{\max}(z^{-1}) \tag{16.156}$$

where it is assumed that the channel gain G is unity. In the time domain Equation (16.156) becomes

$$h^{\text{inv}}(m) = h^{\text{inv}}_{\min}(m) * h^{\text{inv}}_{\max}(m) \tag{16.157}$$

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 $\hat{h}^{\text{inv}}_{\min}(i,m)$, $\hat{h}^{\text{inv}}_{\max}(i,m)$ 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

$$\hat{h}^{\text{inv}}_{\min}(i,0) = \hat{h}^{\text{inv}}_{\max}(i,0) = 1 \tag{16.158}$$

(2) Calculation of the minimum-phase polynomial

$$\hat{h}^{\text{inv}}_{\min}(i,m) = \frac{1}{m}\sum_{k=2}^{m+1} \hat{A}^{(k-1)}\hat{h}^{\text{inv}}_{\min}(i,m-k+1) \quad i=1,\ldots,P_1 \tag{16.159}$$

(3) Calculation of the maximum-phase polynomial

$$\hat{h}^{\text{inv}}_{\max}(i,m) = \frac{1}{m}\sum_{k=m+1}^{0} \hat{B}^{(1-k)}\hat{h}^{\text{inv}}_{\max}(i,m-k+1) \quad i=-1,\ldots,-P_2 \tag{16.160}$$

The maximum-phase and minimum-phase components of the inverse channel response are combined in Equation (16.157) to give the inverse channel equaliser.

16.7 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 non-invertibility 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 telecommunication the restoration of both the amplitude and the timing of the transmitted symbols are of interest, and hence we need to equalise for both the magnitude and the phase distortions.

In Section 16.1, 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 16.2, 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 16.3, 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 16.4, we considered Bayesian equalisation methods, and studied the case where the channel input is modelled by a set of hidden Markov models. Section 16.5 introduced channel equalisation methods for removal of inter-symbol interference in digital telecommunication systems, and finally in Section 16.6, we considered the use of higher-order spectra for equalisation of non-minimum-phase channels.

Bibliography

Benveniste A., Goursat M. and Ruget G. (1980) Robust Identification of a Non-minimum Phase System: Blind Adjustment of Linear Equalizer in Data Communications. *IEEE Trans, Automatic Control*, **AC-25**: 385–399.

Bellini S. (1986) Bussgang Techniques for Blind Equalization. *IEEE GLOBECOM Conf. Rec*.: 1634–1640.

Bellini S. and Rocca F. (1988) Near Optimal Blind Deconvolution. *IEEE Proc. Int. Conf. Acoustics, Speech, and Signal Processing*. **ICASSP-88**: 2236–2239.

Belfiore C.A. and Park J.H. (1979) Decision Feedback Equalization. *Proc. IEEE*, **67**: 1143–1156.

Gersho A. (1969) Adaptive Equalization of Highly Dispersive Channels for Data Transmission. *Bell System Tech. J*., **48**: 55–70.

Godard D.N. (1974) Channel Equalization using a Kalman 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*. Comm., 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 Int. Conf. Acoustics, Speech and Signal Processing*: 79–82.

Hariharan S. and Clark A.P. (1990) HF Channel Estimation using a Fast Transversal Filter Algorithm. *IEEE Trans. Acoustics, Speech and Signal Processing*, **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. *Int. Conf. on Spoken Language Processing* **Tu.fPM.1**.1: 85–88.

Lucky R.W. (1965) Automatic Equalization of Digital Communications. *Bell System Tech. J*., **44**: 547–588.

Lucky R.W. (1965) Techniques for Adaptive Equalization of Digital Communication Systems. *Bell System Tech. J*., **45**: 255–286.

Mendel J.M. (1990) *Maximum Likelihood Deconvolution: A Journey into Model Based Signal Processing*. Springer-Verlag, 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 Jouvet D. (1993) On-Line Adaptation of A Speech Recogniser to Variations in Telephone Line Conditions, *Proc. 3rd European Conf. On Speech Communication and Technology*. EuroSpeech-**93**(2): 1247–1250.

Monsen P. (1971) Feedback Equalization for Fading Dispersive Channels. IEEE Trans. Information 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. Acoustics, Speech and Signal Processing*, **ASSP-36**: 1911–1913.

Nowlan S.J., Hinton G.E. (1993) A Soft Decision-Directed Algorithm for Blind Equalization *IEEE Trans. Communications*, **41**(2): 275–279.

Pan R., Nikias C.L. (1988) Complex Cepstrum of Higher Order Cumulants and Non-minimum Phase Identification. *IEEE Trans. Acoustics, Speech and Signal Processing*, **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.

Raghuveer M.R. and Nikias C.L. (1985) Bispectrum Estimation: A Parameteric Approach. *IEEE Trans. Acoustics, Speech, and Signal Processing*, **ASSP-33**(5): 35–48.

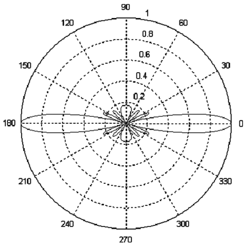
Rosenblatt M. (1985) *Stationary Sequences and Random Fields*. Birkhauser, Boston, MA.

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.

Stockham T.G. and Cannon T.M., Ingebretsen R.B (1975) Blind Deconvolution Through Digital Signal Processing. *IEEE Proc*., **63**(4): 678–692.

Qureshi S.U. (1985) Adaptive Equalization. *IEEE Proc*. **73**(9): 1349–1387.

Ungerboeck G. (1972) Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication. *IBM J. Res. Dev*., **16**: 546–555.



17

Speech Enhancement: Noise Reduction, Bandwidth Extension and Packet Replacement

Speech enhancement systems aim to improve the quality and intelligibility of speech and reduce communication fatigue due to noise, loss of speech packets or limited bandwidth.

Speech enhancement benefits a wide range of applications such as wifi and cellular mobile phones, VoIP, hands-free phones, teleconferencing, in-vehicle cabin communication, hearing aids, automated voice services based on speech recognition and synthesis, speech forensics and reconstruction of old archived records.

This chapter provides an overview of the main speech enhancement methods for single-input and multiple-input noise reduction, bandwidth extension and lost packet replacement.

Single-input noise reduction systems strive to suppress the audibility of the noise by utilising the temporal–spectral structures of signal and noise processes. The speech enhancement is usually achieved through multiplication of the signal frequency components by a gain factor derived from the estimates of the prior and the posterior signal-to-noise ratios.

Multiple-input systems, on the other hand, strive to separate the noise from the noisy speech signal. Examples of multiple-input speech enhancement systems are adaptive noise cancellation, microphone array beam formers and independent component analysis.

Conventional telephony speech is narrowband as it is limited to a bandwidth of 300 Hz to 3400 Hz whereas wideband broadcast speech have a bandwidth of 20 Hz to 20 000 Hz. We will study signal processing methods for extrapolation of bandwidth of narrowband telephony speech to wideband broadcast quality speech.

A major problem with packet transmission of voice over internet (VoIP) is loss of packets due to queuing or connection loss. In this chapter we extend the interpolation methods described in Chapter 11 to consider a speech model-based method for interpolation or extrapolation of lost speech packets.

Finally we consider several commonly used speech distortion measures for evaluation of the gain of speech enhancement methods.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi

17.1 An Overview of Speech Enhancement in Noise

The simplest form of noise reduction is spectral subtraction which subtracts 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 it often introduces artifacts and distortions.

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. Wiener filter requires estimates of the power spectra, or equivalently the correlation matrices, of speech and noise.

High performance speech enhancement methods are based on Bayesian estimation methods requiring estimates of the parameters of the functions that describe the likelihood and the prior distributions of the signal and noise processes. The Bayesian speech enhancement methods include Kalman filters, minimum mean squared error (MMSE), maximum a posteriori (MAP) and ultimately the signal restoration methods that are based on hidden Markov models (HMMs) of speech and noise.

Figure 17.1 illustrates a classification of the main signal processing methods for enhancement of noisy speech into the following two broad types:

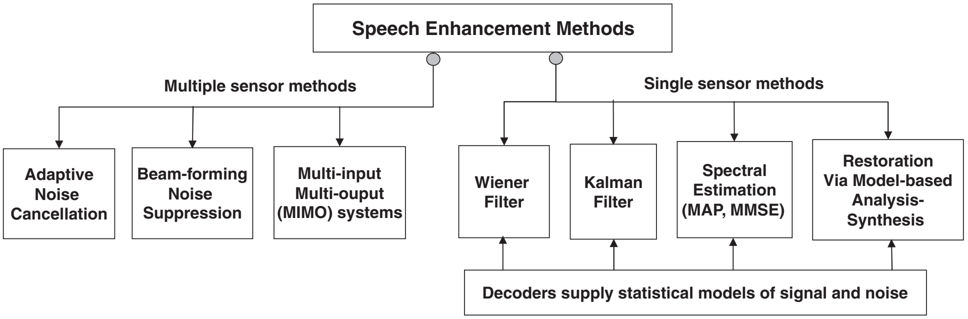

Speech Enhancement Methods
Multiple sensor methods
Single sensor methods
Adaptive Noise Cancellation
Beam-forming Noise Suppression
Multi-input Multi-ouput (MIMO) systems
Wiener Filter
Kalman Filter
Spectral Estimation (MAP, MMSE)
Restoration Via Model-based Analysis-Synthesis
Decoders supply statistical models of signal and noise

Figure 17.1 A categorisation of speech enhancement methods. Note that statistical models can optionally provide single-input noise reduction methods with the additional information needed for improved performance.

(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 ratios of the frequency spectrum of the input signal. Single-input systems rely on the statistical models of speech and noise, which may be estimated on-line 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-microphone 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-input 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 spatially configured and adapted for optimum performance. Multiple-input noise reduction systems are useful for teleconference systems and for in-car cabin communication systems.

Note that 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-input noise cancellation systems are combined.

17.2 Single-Input Speech Enhancement Methods

In single-input systems the only available signal is the noisy speech; in addition statistical models of speech or noise may be available. In applications where speech enhancement and recognition are performed in the same system, the results of speech recognition can provide the speech enhancement method with such statistical information as the power spectra or correlation matrices of the desired signal obtained from decoding the most-likely speech and noise models. Single-input noise reduction methods include Wiener filter, spectral subtraction, Kalman filter, the MMSE noise suppression method and speech restoration via model-based analysis and synthesis methods as described in this section.

17.2.1 Elements of Single-Input Speech Enhancement

Assuming that the speech signal $x(m)$ and the noise $n(m)$ are additive the noisy speech $y(m)$ is modelled as

$$y(m) = x(m) + n(m) \tag{17.1}$$

where the integer variable m denotes the discrete-time index. It is generally assumed that the speech signal is uncorrelated with noise; this is a reasonable assumption since in most cases the signal and noise are generated by independent sources.

The general form of a typical speech enhancement method is shown in Figure 17.2. The speech enhancement system is composed of a combination of the following modules:

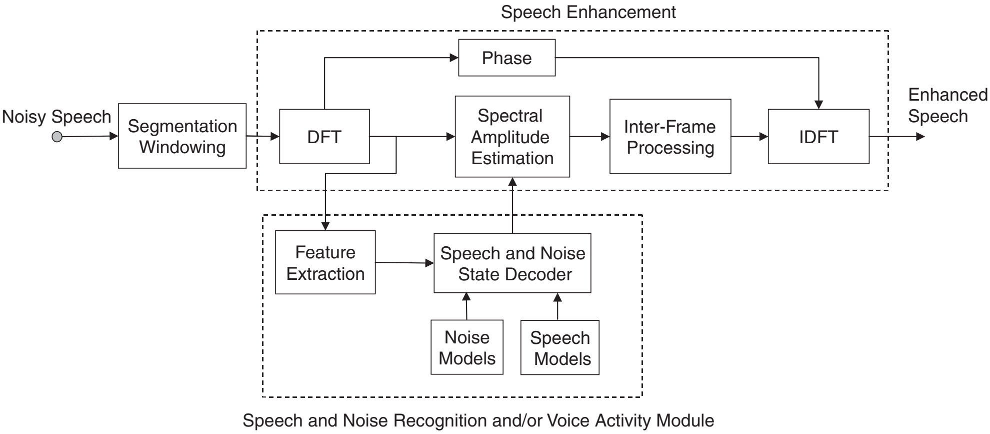


Figure 17.2 Block diagram illustration of a speech enhancement system.

(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 Hamming, Hanning or Hann windows.
(2) Discrete Fourier transformation (DFT) 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, supply the speech estimator with the required statistics (power spectra, correlation matrices, etc.) of speech and noise.
(6) Voice activity detection (integrated with, or substituting, speech and noise recognition) is used to estimated 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.

17.2.1.1 Segmentation and Windowing of Speech Signals

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. The signal within each frame is assumed to be a stationary process. The choice of the length of speech frames (typically set to 20–30 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. Note that with a window length of N samples and a sampling rate of F_s Hz the frequency resolution of DFT is F_s/N Hz.

17.2.1.2 Spectral Representation of Speech and Noise

Speech is segmented into overlapping frames of N samples and transformed to the frequency domain via discrete Fourier transform (DFT). In the frequency domain the noisy speech samples can be represented as

$$Y(k) = X(k) + N(k) \qquad k = 0, \ldots, N-1 \tag{17.2}$$

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/sec) or kF_s/N (Hz) where F_s are the sampling frequencies.

Equation (17.2) can be written in complex polar form in terms of the magnitudes and the phases of the signal and noise at discrete frequency k as

$$Y_k e^{j\theta_{Y_k}} = X_k e^{j\theta_{X_k}} + N_k e^{j\theta_{N_k}} \qquad k = 0, \ldots, N-1 \tag{17.3}$$

where $Y_k = |Y(k)|$ and $\theta_{Y_k} = \tan^{-1}\big(\mathrm{Im}(Y(k))/\mathrm{Re}\big(Y(k)\big)\big)$ are the magnitude and phase of the frequency spectrum of $Y(k)$ 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 in critical bands of the auditory spectral analysis systems.

17.2.1.3 Linear Prediction Model Representation of Speech and Noise

The correlation of speech (or noise) samples can be modelled with a linear prediction (LP) model, as introduced in Chapter 8. Using a linear prediction model of speech and noise, the noisy speech is expressed as

$$y(m) = \underbrace{\sum_{k=1}^{P} a_k x(m-k) + e(m)}_{\text{LP model of Speech}} + \underbrace{\sum_{k=1}^{Q} b_k n(m-k) + v(m)}_{\text{LP model of Noise}} \tag{17.4}$$

where a_k and b_k are the coefficients of LP models of speech and noise, $e(m)$ and $v(m)$ are the random inputs of the LP models and P and Q are the model orders 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.

17.2.1.4 Inter-Frame and Intra-Frame Correlations

The two main issues in modelling noisy speech are the following:

(1) Modelling and utilisation of the probability distributions and the intra-frame correlations of speech and noise samples *within* each noisy speech frame of N samples.
(2) Modelling and utilisation of the probability distributions and the inter-frame 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 the speech samples $x(m)$ with a sinusoidal basis function $e^{-j2\pi km/N}$. The intra-frame 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 speech enhancement methods based on linear prediction models (LP) of speech, the LP model's poles model the spectral correlations within each frame. However, the de-noising of linear prediction model is achieved through de-noising the discrete samples of the frequency response of noisy speech and that process ignores the correlation of spectral samples. The optimal utilisation of the inter-frame and intra-frame correlations of speech samples is a continuing research issue.

17.2.1.5 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 estimation of clean speech including Wiener filter, spectral subtraction, Kalman filter, the minimum mean squared error (MMSE) and the maximum a posterior (MAP) methods. For the proper functioning of the speech estimation module 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 as explained next.

17.2.1.6 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 time-varying statistical parameters (and in particular the power spectra or equivalently the correlation matrices) of the speech and noise processes. An estimate of the noise statistics can be obtained from the speech-inactive periods, however for the best results the speech and noise statistical parameters 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 (HMMs). Hidden Markov models, or alternatively Gaussian mixture models (GMMs), can also be used for modelling non-stationary 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.

17.2.1.7 Cost of Error Functions in Speech Estimation

The Bayesian cost function, introduced in Chapter 4, provides a general framework for speech estimation and for calculation and minimisation of the cost of estimation error function.

Most speech enhancement methods use a cost function that involves minimising some function of the average squared difference (error) between the discrete-frequency spectrum of clean speech X_k and its estimate $\hat{X}_k$ i.e. $\mathcal{E}\left[(X_k - \hat{X}_k)^2\right]$, where the operation $\mathcal{E}[\cdot]$ denotes expectation or averaging. However, non-linear estimation error functions such as $\mathcal{E}\left[\left(X_k^{\beta_k} - \hat{X}_k^{\beta_k}\right)^{\alpha_k}\right]$ are also used, where varying the parameters β_k and α_k provides a family of estimators; note that for the special case of $\beta_k = 1$ and $\alpha_k = 2$ this function is the squared error function. Some cost of error functions attempt to utilise the knowledge of the non-linear transformations of frequency and amplitude of speech that take place within the cochlea.

17.2.2 Wiener Filter for De-noising Speech

Wiener filter theory, introduced in Chapter 6, forms the foundation of speech de-noising systems. The output of a discrete-time Wiener filter is given by

$$\hat{x}(m) = \sum_{i=0}^{P} w(i)y(m-i) \tag{17.5}$$

where $w(i)$ are the filter coefficients for de-noising the input speech $y(m)$ and $\hat{x}(m)$ is the Wiener estimate of the clean speech signal $x(m)$. The Wiener filter coefficient vector $\boldsymbol{w} = [w(0),\ w(1), \ldots,\ w(P)]^{\mathrm{T}}$ was derived in Chapter 6 as

$$\boldsymbol{w} = \boldsymbol{R}_{yy}{}^{-1}\boldsymbol{r}_{xy} \tag{17.6}$$

where $\boldsymbol{R}_{yy}$ is the autocorrelation matrix of the noisy speech signal $\boldsymbol{y}$ and $\boldsymbol{r}_{yx}$ is the cross-correlation vector of the clean speech $\boldsymbol{x}$ and noisy speech $\boldsymbol{y}$.

For uncorrelated additive speech and noise we have $\boldsymbol{R}_{yy} = \boldsymbol{R}_{xx} + \boldsymbol{R}_{nn}$, hence the Wiener filter Equation (17.6) can be written as

$$\boldsymbol{w} = [\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn}]^{-1}\,\boldsymbol{r}_{xx} \tag{17.7}$$

where $\boldsymbol{R}_{xx}$ and $\boldsymbol{R}_{nn}$ are the autocorrelation matrices of the speech and noise respectively and $\boldsymbol{r}_{xx}$ 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

$$W(k) = \frac{P_{XX}(k)}{P_{XX}(k) + P_{NN}(k)} \tag{17.8}$$

where $W(k)$ is the frequency response of the Wiener filter and $P_{XX}(k)$ and $P_{NN}(k)$ are the power spectra of speech and noise respectively and k is the discrete frequency variable. Figure 17.3 shows a block diagram implementation of a frequency domain Wiener filter.

By dividing the numerator and the denominator of Equation (17.8) by $P_{NN}(k)$, the Wiener filter can be expressed in terms of the signal-to-noise ratio as

$$W(k) = \frac{SNR(k)}{SNR(k) + 1} \tag{17.9}$$

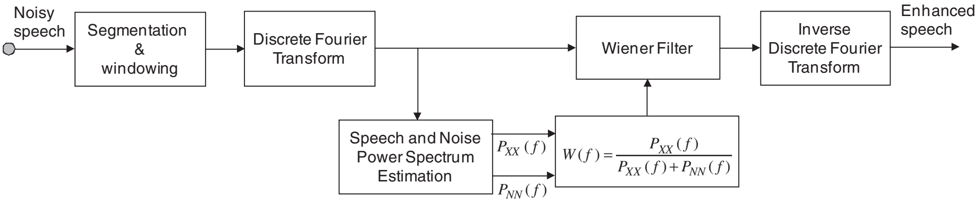

Noisy speech
Segmentation & windowing
Discrete Fourier Transform
Wiener Filter
Inverse Discrete Fourier Transform
Enhanced speech
Speech and Noise Power Spectrum Estimation
$P_{XX}(f)$
$P_{NN}(f)$
$W(f)=\frac{P_{XX}(f)}{P_{XX}(f)+P_{NN}(f)}$

Figure 17.3 Block diagram overview of implementation of a Wiener filter for a speech enhancement system.

Equation (17.9) reveals an important aspect of the general workings of 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.*

17.2.2.1 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

$$W(f)=\frac{P_{XX}(f)}{P_{YY}(f)}=\frac{G_X^2/A_X^2(f)}{G_Y^2/A_Y^2(f)}=\frac{G_X^2}{G_Y^2}\frac{A_Y^2(f)}{A_X^2(f)} \tag{17.10}$$

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 time domain a square root Wiener filter Equation (17.10) can be implemented as

$$\hat{x}(m)=\sum_{k=1}^{P}a_x(k)\hat{x}(m-k)+\frac{G_X}{G_Y}\sum_{k=0}^{Q}a_y(k)y(m-k) \tag{17.11}$$

where $a_x(k)$ and $a_y(k)$ are the coefficients of autoregressive models of clean speech and noisy speech respectively.

17.2.2.2 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 filter of Equation (17.8) or equivalently the estimation of the correlation matrices of speech and noise, $\boldsymbol{R}_{xx}$ and $\boldsymbol{R}_{nn}$, for the time domain Wiener filter of Equation (17.7). This is not a trivial task as speech and most noise processes are non-stationary processes.

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 (HMMs), or Gaussian mixture models (GMMs), of speech and noise using a Viterbi decoder as illustrated in Figure 17.4. 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 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 (a) are used to implement state-based Wiener filters.

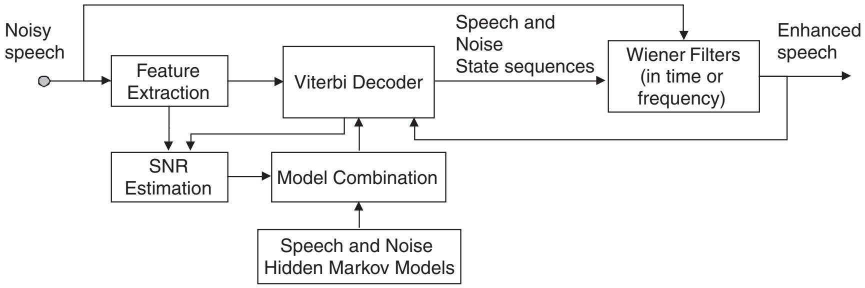

Noisy speech
Feature Extraction
Viterbi Decoder
Speech and Noise State sequences
Wiener Filters (in time or frequency)
Enhanced speech
SNR Estimation
Model Combination
Speech and Noise Hidden Markov Models

Figure 17.4 Block diagram illustration of a speech enhancement system based on Wiener filters and hidden Markov models (HMMs).

In HMM-based Wiener filtering the choice of the speech features 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 the HMM's 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 Equations (17.8) and (17.10).

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

$$\hat{X}(k) = \left[\sum_{\beta=1}^{M}\sum_{\gamma=1}^{L} p(\beta,\gamma)\ W_{\beta,\gamma}(k)\right] Y(k) \tag{17.12}$$

where $p(\beta,\gamma)$ is the estimated probability of speech and noise spectra from mixture γ of HMM state β.

17.2.3 Spectral Subtraction of Noise

A simple and widely studied speech enhancement method is the spectral subtraction method illustrated in Figure 17.5. In spectral subtraction an estimate of the average magnitude spectrum of the noise is subtracted from the magnitude spectrum of noisy speech. The spectral subtraction filter can be expressed as the product of the noisy speech spectrum $Y(k)$ and a spectral gain function $W_{SS}(k)$

$$\hat{X}(k) = W_{SS}(k)Y(k) \tag{17.13}$$

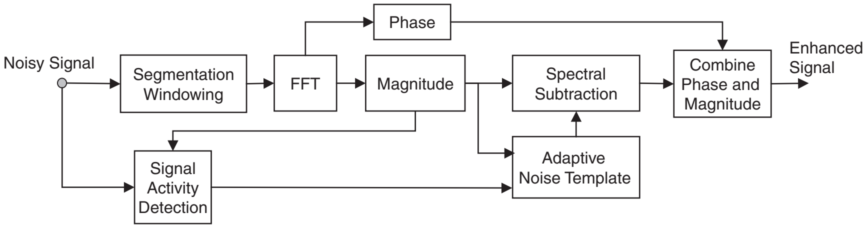


Figure 17.5 Block diagram illustration of an FFT-based spectral subtraction system for de-noising speech.

where the frequency response of the spectral subtraction filter $W_{SS}(k)$ is

$$W_{\text{SS}}(k) = fn\left(1 - \frac{\alpha(k)\hat{N}(k)}{Y(k)}\right) \tag{17.14}$$

where $\hat{N}(k)$ is an estimate of the noise average amplitude spectrum, $\alpha(k)$ is a frequency-dependent subtraction factor and the function $fn(\cdot)$ is chosen to avoid negative values of $W_{\text{SS}}(k)$ and provide a smoother frequency response when the signal-to-noise ratio drops to relatively lower values. The form of the function $fn(\cdot)$ can be chosen as

$$W_{\text{SS}}(k) = \begin{cases} 1 - \alpha(k)\hat{N}(k)/Y(k) & \text{if } SNR(k) > SNR_{\text{Thresh}} \\ \gamma \exp\left(-\beta\left(SNR_{\text{Thresh}} - SNR(k)\right)\right) & \text{else} \end{cases} \tag{17.15}$$

where 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 known as musical noise. The shortcomings of the spectral subtraction method can be summarised as follows:

(1) The only statistic 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 was 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.

17.2.3.1 Spectral Subtraction Using LP Model Frequency Response

Spectral subtraction can be applied either on the short-time spectral amplitude (STSA) of noisy speech obtained from DFT or on the magnitude of the frequency response of a linear prediction (LP) model of noisy speech, as illustrated in Figure 17.6.

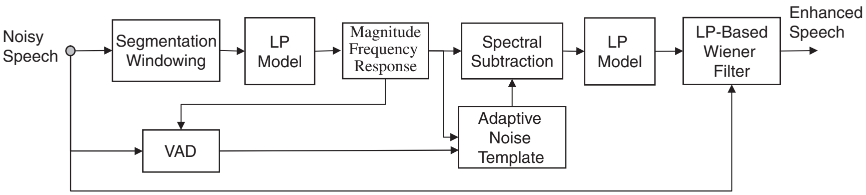


Figure 17.6 Block diagram illustration of LP-based spectral subtraction system for de-noising speech. Note that the LP model following the spectral subtraction module is intended to smooth the output of spectral subtraction.

For LP-based spectral subtraction (LPSS), the filter response $W_{\text{LPSS}}(k)$ is obtained from equations similar to (17.14) and (17.15) with the main difference that instead of the DFT-based power-spectrum, the LP power-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 $W_{\mathrm{LPSS}}(k)$ from Equation (17.14).
(3) Restore the speech signal through application of the smoothed LP-spectral subtraction filter to the noisy speech.

17.2.4 Bayesian MMSE Speech Enhancement

The probabilistic minimum mean squared error estimation (MMSE) of the short-time spectral amplitude (STSA) of speech is a Bayesian estimation method with a mean squared error cost function. The Bayesian MMSE estimation of spectral amplitude is obtained as the posterior mean of the signal as

$$\hat{X}_k = \frac{\int\limits_{-\infty}^{\infty}\int\limits_{0}^{2\pi} X_k\, p\left(Y(k)|X_k,\theta_{X_k}\right) P\left(X_k,\theta_{X_k}\right) d\theta_{X_k}\; dX_k}{\int\limits_{-\infty}^{\infty}\int\limits_{0}^{2\pi} p\left(Y(k)|X_k,\theta_{X_k}\right) d\theta_{X_k} dX_k} \tag{17.16}$$

where $p(X(k)|Y_k,\theta_{Xk}) = p(Y(k)|X_k,\theta_{X_k})\; p(X_k,\theta_{X_k})/p(Y_k,\theta_{X_k})$ is the posterior probability of clean speech $X(k)$ given noisy observation $Y(k)$. The MMSE equation (17.16) requires the likelihood of the noisy speech $p\left(Y(k)|X_k,\theta_{X_k}\right)$ and the prior probability density function of the clean speech $p\left(X_k,\theta_{X_k}\right)$.

Ephraim and Malah (1985) derived a MMSE spectral amplitude estimation algorithm known as the Ephraim–Malah suppression rule. They used a Gaussian distribution for clean speech (which results in a Rayleigh distribution for the magnitude spectrum of 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:

$$\hat{X}_k = W_{\mathrm{MMSE}}(k) Y_k \tag{17.17}$$

where the gain factor $W_{\mathrm{MMSE}}(k)$ is given by

$$W_{\mathrm{MMSE}}(k) = \Gamma(1.5)\frac{\sqrt{v_k}}{\gamma_k}\exp\left(-\frac{v_k}{2}\right)\left[(1+v_k)I_0\left(\frac{v_k}{2}\right)+v_k I_1\left(\frac{v_k}{2}\right)\right] \tag{17.18}$$

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

$$v_k = \frac{\xi(k)}{1+\ \xi(k)}\ \gamma_k, \xi_k = \frac{\sigma_X^2(k)}{\sigma_N^2(k)},\ \gamma_k = \frac{Y^2(k)}{\sigma_N^2(k)} \tag{17.19}$$

where $\sigma_X^2(k)$ and $\sigma_N^2(k)$ are the variances 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 17.7 shows a comparison of the performance of spectral subtraction based on linear prediction model of speech (LPSS) and the MMSE method for de-noising of speech observed in train noise (Yan and Vaseghi, 2003).

17.2.5 Kalman Filter for Speech Enhancement

Kalman filter, described in Chapter 7, differs from Wiener filter in several ways, the most important of which is that the Kalman formulation permits explicit inclusion of time-varying equations that model the

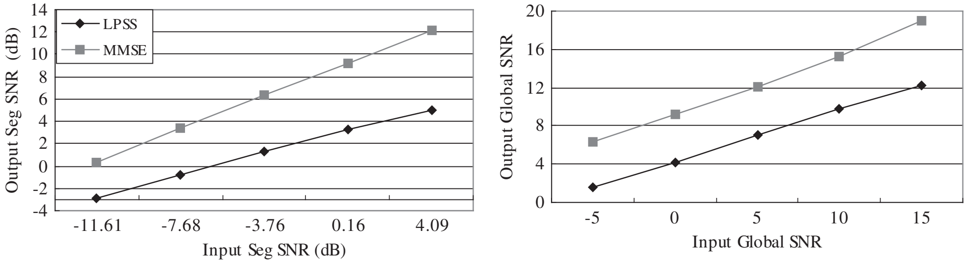


Figure 17.7 Performance comparison of LPSS and MMSE: output SNR vs input SNR.

dynamics of speech and noise process. For de-noising speech, Kalman filter can be implemented in the time or frequency domain. The conventional method is time-domain Kalman filter.

17.2.5.1 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

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + e(m) \tag{17.20}$$

$$n(m) = \sum_{k=1}^{Q} b_k n(m-k) + v(m) \tag{17.21}$$

where a_k and b_k are the coefficients of P^{th} order and Q^{th} order autoregressive models of speech and noise respectively. Equations (17.20) and (17.21) can be expressed in a state-space form for Kalman filtering as

$$\begin{cases} \boldsymbol{x}(m) = \boldsymbol{A}_x \boldsymbol{x}(m-1) + \boldsymbol{g}_x e(m) & (17.22) \\ x(m) = \boldsymbol{h}_x \boldsymbol{x}(m) & (17.23) \end{cases}$$

$$\begin{cases} \boldsymbol{n}(m) = \boldsymbol{A}_n \boldsymbol{n}(m-1) + \boldsymbol{g}_n v(m) & (17.24) \\ n(m) = \boldsymbol{h}_n \boldsymbol{n}(m) & (17.25) \end{cases}$$

where the AR model coefficient matrices of speech and noise are defined as

$$\boldsymbol{A}_x = \begin{bmatrix} a_1 & a_2 & \cdots & a_{P-1} & a_P \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{A}_n = \begin{bmatrix} b_1 & b_2 & \cdots & b_{Q-1} & b_Q \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{17.26}$$

The signal and noise vectors are $\boldsymbol{x}(m) = [x(m),\ x(m-1), \ldots,\ x(m-P)]^{\mathrm{T}}$, $\boldsymbol{n}(m) = [n(m),\ n(m-1), \ldots,\ n(m-Q)]^{\mathrm{T}}$ and $\boldsymbol{h}_x = \boldsymbol{g}_x = [1,\ 0, \ldots,\ 0]^{\mathrm{T}}$ are P-dimensional vectors and $\boldsymbol{h}_n = \boldsymbol{g}_n = [1,\ 0, \ldots,\ 0]^{\mathrm{T}}$ are Q-dimensional vectors.

Signal and noise process equations

The signal and noise processes can be arranged into an augmented vector process equation as

$$\underbrace{\begin{bmatrix} \boldsymbol{x}(m) \\ \boldsymbol{n}(m) \end{bmatrix}}_{\substack{\boldsymbol{z}(m) \\ (P+Q)\times 1}} = \underbrace{\begin{bmatrix} \boldsymbol{A}_x & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_n \end{bmatrix}}_{\substack{\boldsymbol{A} \\ (P+Q)\times(P+Q)}} \underbrace{\begin{bmatrix} \boldsymbol{x}(m-1) \\ \boldsymbol{n}(m-1) \end{bmatrix}}_{\substack{\boldsymbol{z}(m-1) \\ (P+Q)\times 1}} + \underbrace{\begin{bmatrix} \boldsymbol{g}_x & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{g}_n \end{bmatrix}}_{\substack{\boldsymbol{G} \\ (P+Q)\times 2}} \underbrace{\begin{bmatrix} e(m) \\ v(m) \end{bmatrix}}_{\substack{\boldsymbol{w}(m) \\ 2\times 1}} \tag{17.27}$$

Equation (17.27) shows the dimensions of each vector and matrix. As indicated underneath the terms in Equation (17.27), in a compact notation the state process equation can be written as

$$\boldsymbol{z}(m) = \boldsymbol{A}\,\boldsymbol{z}(m-1) + \boldsymbol{G}\,\boldsymbol{w}(m) \tag{17.28}$$

Note that within the Kalman methodology an estimate of the state Equation (17.28) provides a decomposition of the signal and noise processes.

Noisy observation (measurement) equation

The noisy observation, i.e. the sum of speech and noise signals, may be expressed as

$$y(m) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{z}(m) = x(m) + n(m) \tag{17.29}$$

where

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_n \end{bmatrix} \tag{17.30}$$

and $\boldsymbol{z}(m) = [\boldsymbol{x}(m)\ \boldsymbol{n}(m)]^{\mathrm{T}}$ as defined in Equation (17.27). Note that in this formulation of the Kalman filter, the signal and noise form the state process vector in Equation (17.27) and the noisy observation, given by Equation (17.29), is a linear transformation of the state vector that simply adds signal and noise. The Kalman filter equations given in Chapter 7 and adapted here are as follows.

State vector prediction equation

$$\hat{\boldsymbol{z}}(m|m-1) = \boldsymbol{A}(m-1)\,\hat{\boldsymbol{z}}(m-1) \tag{17.31}$$

Covariance matrix of prediction error

$$\boldsymbol{P}(m|m-1) = \boldsymbol{A}(m-1)\boldsymbol{P}(m-1)\boldsymbol{A}^{\mathrm{T}}(m-1) + \boldsymbol{G}\boldsymbol{W}(m-1)\boldsymbol{G} \tag{17.32}$$

$\boldsymbol{W} = diag[\sigma_e^2, \sigma_v^2]$ is the covariance matrix of the vector $\boldsymbol{w}$ in Equations (17.27)–(17.28).

Kalman gain vector

$$\boldsymbol{K}(m) = \boldsymbol{P}(m|m-1)\boldsymbol{h}\left(\boldsymbol{h}^{\mathrm{T}}\boldsymbol{P}(m|m-1)\,\boldsymbol{h}\right)^{-1} \tag{17.33}$$

State update estimate equation

$$\hat{\boldsymbol{z}}(m) = \hat{\boldsymbol{z}}(m|m-1) + \boldsymbol{K}(m)\left(y(m) - \boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{z}}(m|m-1)\right) \tag{17.34}$$

Note that the innovation signal $y(m) - \boldsymbol{h}^{\mathrm{T}}\hat{\boldsymbol{z}}(m|m-1)$ is a mixture of the unpredictable parts of the signal and the noise. The covariance matrix of estimation error is given by

$$\boldsymbol{P}(m) = \left(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{h}^{\mathrm{T}}\right)\boldsymbol{P}(m|m-1) \tag{17.35}$$

The application of Kalman filter requires estimates of the AR models of speech and noise $\boldsymbol{a} = [a_1, a_2, \ldots, a_P]$ and $\boldsymbol{b} = [b_1, b_2, \ldots, b_Q]$, Figure 17.8. These are obtained from an application of the estimate–maximize (EM) algorithm, described in Chapter 7, which effectively yields the following normal equation (Yule –Walker equation) as

$$\hat{\boldsymbol{a}} = \boldsymbol{R}_{\hat{x}\hat{x}}{}^{-1} r_{\hat{x}\hat{x}} \tag{17.36}$$

$$\hat{\boldsymbol{b}} = \boldsymbol{R}_{\hat{n}\hat{n}}{}^{-1} \boldsymbol{r}_{\hat{n}\hat{n}} \tag{17.37}$$

where the autocorrelation matrices of speech and noise, $\boldsymbol{R}_{\hat{x}\hat{x}}$ and $\boldsymbol{R}_{\hat{n}\hat{n}}$, are obtained from Kalman estimates as

$$\boldsymbol{R}_{\hat{x}\hat{x}} = \mathcal{E}\left(\left[\hat{x}(m), \ldots, \hat{x}(m-P)\right]^{\mathrm{T}} \left[\hat{x}(m), \ldots, \hat{x}(m-P)\right]\right) \tag{17.38}$$

$$\boldsymbol{R}_{\hat{n}\hat{n}} = \mathcal{E}\left(\left[\hat{n}(m), \ldots, \hat{n}(m-Q)\right]^{\mathrm{T}} \left[\hat{n}(m), \ldots, \hat{n}(m-Q)\right]\right) \tag{17.39}$$

Figure 17.8 is an outline illustration of the Kalman filter method of enhancement of noisy speech.

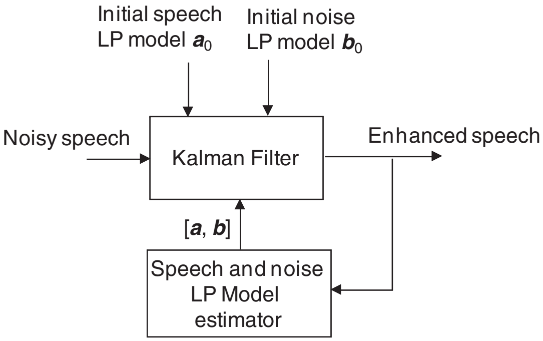


Figure 17.8 Illustration of Kalman filter method of enhancement of noisy speech.

17.2.6 Speech Enhancement Using LP-HNM Model

Linear prediction (LP) and harmonic noise models (HNM) are the two main methods for modelling and synthesis of speech waveforms. LP and HNM offer complementary advantages; an LP model provides a good fit for the spectral envelope of speech whereas an HNM is good at modelling the details of the harmonic plus noise structures of speech. These two methods can be combined for speech enhancement.

The temporal trajectory of the estimates of the parameters of LP and HNM models can be tracked with Viterbi classifiers, described in Chapter 5, and smoothed with Kalman filters. For noisy speech processing this is a different approach to spectral amplitude estimation methods which generally model each individual spectral sample in isolation without fully utilising the information on the wider spectral–temporal structures that may be used to good effect in the de-noising process to obtain improved results.

In this subsection a speech enhancement method is described based on a formant-tracking LP (FTLP) model of the spectral envelope and an HNM model of the excitation. The FTLP model obtains enhanced estimates of the LP parameters of speech along the formant trajectories. Formants are the resonances of the vocal tract and their trajectories describe the contours of energy concentrations in time and frequency. Although formants are mainly defined for voiced speech, characteristic energy concentration contours also exist for unvoiced speech at relatively higher frequencies.

17.2.6.1 Overview of LP-HNM Enhancement System

The LP-HNM speech enhancement method is illustrated in Figure 17.9 and consists of the following sections:

(1) A pre-cleaning module, composed of a spectral amplitude estimator, for de-noising speech spectrum prior to estimation of the LP model and formant parameters.
(2) A formant-tracking method incorporating Viterbi decoders and Kalman filters for tracking and smoothing the temporal trajectories of the estimates of the formants and poles of the LP model.
(3) A pitch extraction method incorporating Viterbi decoders and Kalman filters for pitch smoothing.
(4) A method for estimation of a harmonic noise model of clean excitation with Kalman filters used for modelling and de-noising the temporal trajectory of the noisy excitation.

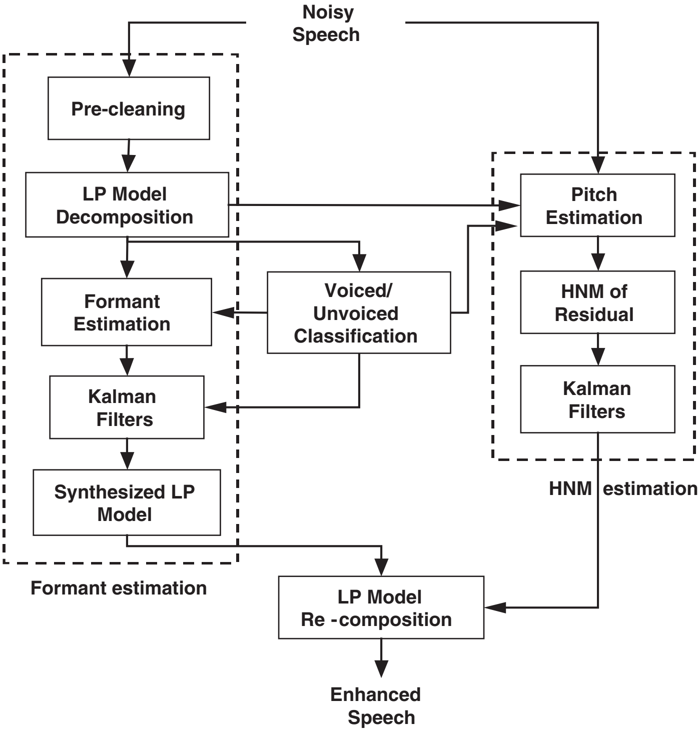


Figure 17.9 Overview of the FTLP-HNM model for enhancement of noisy speech.

The LP model of speech $X(z,m)$ may be expressed as

$$X(z,m) = E(z,m)V(z,m) \tag{17.40}$$

where $E(z,m)$ is the z-transform of the excitation signal and $V(z,m)$, the z-transform of an LP model of the spectral envelope of speech, can be expressed as

$$V(z,m) = G(m)\frac{1}{1+r_0(m)z^{-1}}\prod_{k=1}^{P/2}\frac{1}{1-2r_k(m)\cos\left(\varphi_k(m)\right)z^{-1}+r_k^2(m)z^{-2}} \tag{17.41}$$

where $r_k(m)$ and $\varphi_k(m)$ are the time-varying radii and the angular frequencies of the poles of the LP model respectively, $P+1$ is the LP model order and $G(m)$ is the gain.

The speech excitation can be modelled as a combination of the harmonic and the noise contents of the excitation as

$$E(f,m)=\sum_{k=1}^{L(m)}A_k(m)G\big(f-kF_0(m)+\Delta_k(f,m)\big)+V(f,m) \tag{17.42}$$

where f is the frequency variable, $L(m)$ denotes the number of harmonics, $F_0(m)$ is the time-varying fundamental frequency, $\Delta_k(f,m)$ is the deviation of the k^{th} harmonic from the nominal value of kF_0, $A_k(m)$ are the complex amplitudes of excitation harmonics, $G(f)$ is a Gaussian-shaped model of harmonic shape, and $V(f,m)$ is the noise part of the excitation. The harmonic shape function $G(f)$ has a frequency support equal to F_0 and is selected to model the shape of the harmonics of speech in the frequency domain.

17.2.6.2 Formant Estimation from Noisy Speech

In this section a robust formant-tracking LP model is described composed of pre-cleaning of noisy speech spectrum followed by formant track estimation and Kalman smoothing of formant tracks.

17.2.6.3 Initial-Cleaning of Noisy Speech

Before formant estimation, the noisy speech spectrum is pre-cleaned using the MMSE spectral amplitude estimation method described in Section 17.2.4. After pre-cleaning, the spectral amplitude of speech is converted to a correlation function from which an initial estimate of the LP model of speech is obtained using the Levinson–Durbin method. A formant tracker is then used to process the poles, or equivalently the line spectral frequency (LSF) parameters, of the LP model and obtain an improved estimate of the LP model parameters as described next.

17.2.6.4 Formant Tracking

The poles of the LP model of the pre-cleaned speech are the formant candidates represented by formant feature vectors $\boldsymbol{v}_k$ comprising the frequency F_k, bandwidth B_k and magnitude M_k of the resonance at formants together with their velocity derivatives as

$$\boldsymbol{v}_k=[F_k,B_k,M_k,\Delta F_k,\Delta B_k,\Delta M_k] \qquad k=1,\ldots,N \tag{17.43}$$

where the number of formants is typically set to $N=5$. Velocity derivatives, denoted by Δ, are computed as the slopes of the formant features over time. The probability distributions of formants can be modelled by Gaussian mixture models (GMMs) or HMMs, as described in Chapter 5. A Viterbi classifier is used to classify and track the poles of the LP model and label the poles with different formants. Kalman filters are subsequently employed to smooth formant trajectories. Note that instead of formants one can employ the line spectral frequencies.

For assessment of the performance of the formant tracker, the Wall Street Journal speech database is used to investigate the effect of noise on the estimates of formants (Yan and Vaseghi 2003). The speech examples are degraded by moving car noise or moving train noise with an average SNR in the range 0 to 20 dB. To quantify the contamination of formants by noise a local formant signal-to-noise ratio measure (FSNR) is defined as

$$FSNR(k)=10\ \log\left[\sum_{l\in\ (F_k\pm B_k/2)}X_l^2\Big/\sum_{l\in(F_k\pm B_k/2)}N_l^2\right] \tag{17.44}$$

where X_l and N_l are the magnitude spectra of speech and noise respectively and F_k and B_k are the frequency and bandwidth of the resonance at the k^{th} formant. Figure 17.10 displays the FSNRs of noisy speech in moving car and moving train environments. It is evident that at the signal-to-noise ratio at formants, the FSNRs are higher than the average SNR.

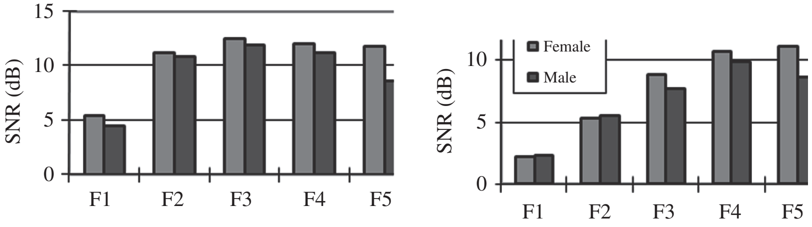


Figure 17.10 Variation of speech SNR at different formants in car noise (left) train noise (right) at average SNR = 0 dB.

To quantify the effects of the noise on formant estimation, an average formant track error measure is defined as

$$E_k = \frac{1}{L}\sum_{m=1}^{L}\left[\left|F_k(m)-\hat{F}_k(m)\right|\Big/F_k(m)\right]\times 100\,\% \qquad K=1,\ldots,N \tag{17.45}$$

where $F_k(m)$ and $\hat{F}_k(m)$ are the formant tracks of clean and noisy speech respectively, m is frame index and L is the number of speech frames over which the error is measured.

Figure 17.11 shows the improvement in formant estimation resulting from pre-cleaning of speech spectra followed by LP analysis and the subsequent enhancement of the poles of the LP model using a Viterbi classifier and Kalman filters. The reference formant tracks, used for calculation of the estimation errors, are obtained from HMMs of formants of clean speech (Yan and Vaseghi 2003). It can be seen that the application of MMSE noise suppression results in a significant reduction of formant tracking error. Further improvement is obtained through the application of Kalman filtering.

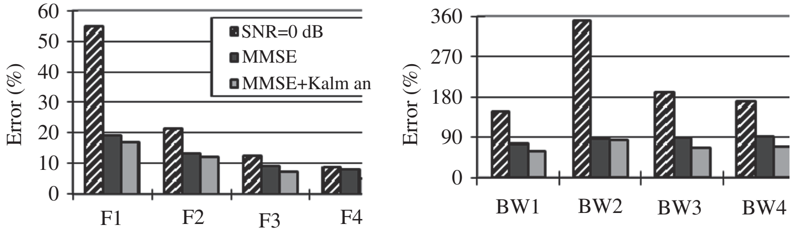


Figure 17.11 Average % error of formant tracks (frequency F_k and bandwidth BW_k) in train noise and cleaned speech using MMSE and Kalman filters; the results were averaged over five males.

17.2.6.5 Harmonic Plus Noise Model (HNM) of Speech Excitation

The harmonic plus noise model (HNM) of speech (Zavarehei 2007, 2008) can be applied for modelling of noisy excitation. The procedure includes the following steps:

(1) Voiced/unvoiced classification of speech.
(2) Estimation and smoothing of the trajectories of the fundamental frequency and harmonic tracks.

(3) Estimation and smoothing of the trajectories of the amplitudes of harmonics.
(4) Estimation of the harmonicity (i.e. proportion of harmonic energy to fricative or aspiration noise) values.
(5) Estimation of the noise component of the excitation.

The estimation of HNM parameters is discussed next.

17.2.6.6 Fundamental Frequency Estimation

Traditionally pitch is derived as the inverse of the time interval τ corresponding to the second largest peak of the autocorrelation of speech. Since autocorrelation of a periodic signal is itself periodic, all the periodic peaks of the autocorrelation can be used in the pitch estimation process. The proposed pitch estimation method is an extension of the autocorrelation-based method to the frequency domain. A pitch estimation error is defined as

$$E(F_0) \; = E - F_0 \sum_{k=1}^{MaxF} \sum_{l=kF_0-M}^{kF_0+M} W(l) \log \; |X(l)| \tag{17.46}$$

where $X(l)$ is the DFT of speech, F_0 is a proposed value of the fundamental frequency (pitch) variable, E is sum of log spectral energy, and $2M+1$ is a band of values about each harmonic frequency. The weighting function $W(l)$ is a SNR-dependent Wiener-type weight. Figure 17.12 provides a comparative illustration of the performance of the proposed pitch estimation method with Griffin's method (1988), at different SNRs for car noise and train noise. It can be seen that the proposed frequency method with SNR weighting provides improved performances in all cases evaluated.

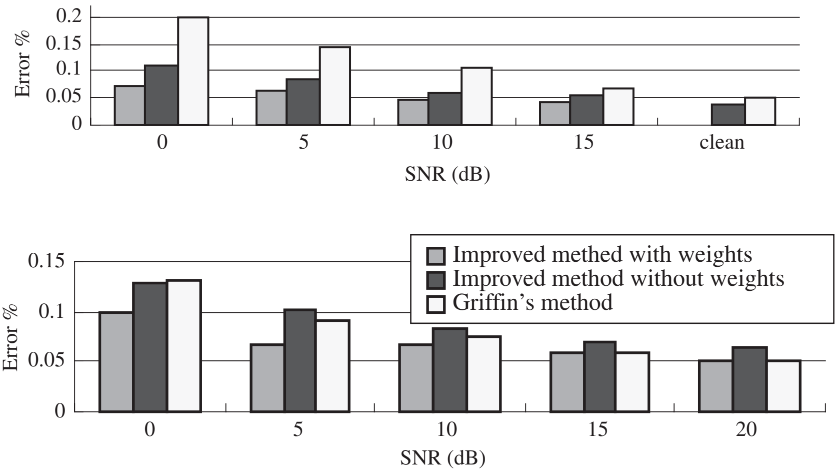


Figure 17.12 Comparison of different pitch track methods for speech in train noise (top) and car noise (bottom) from 0 dB SNR to clean.

17.2.6.7 Estimation of Amplitudes Harmonics of HNM

The harmonic content of speech excitation is modelled as

$$e_h(m) = \sum_{k=1}^{L(m)} A_k(m) \cos\Big(2\pi \big(kF_0(m) + \Delta_k(m)\big)m + \varphi_k(m)\Big) = \boldsymbol{A}^T \boldsymbol{S} \tag{17.47}$$

where $L(m)$ denotes the number of harmonics and $F_0(m)$ is the pitch, $\boldsymbol{A}$ and $\boldsymbol{S}$ are the harmonic amplitude vector and the harmonically related sinusoids vector respectively. Given the harmonics frequencies, the amplitudes $\boldsymbol{A}$ can be obtained either from searching for the peaks of the speech DFT spectrum or through a least square error estimation. The maximum significant harmonic number is obtained from the ability of the harmonic model to synthesise speech locally at the higher harmonics of the pitch. Note that the deviation of the harmonic frequencies, $\Delta_k(m)$, from the nominal value can be obtained from a search for the peak amplitudes about the nominal harmonic frequencies.

The estimate of the amplitudes of clean excitation harmonics is obtained from a set of Kalman filters, one for each harmonic. The Kalman filter is the preferred method here as it models the trajectory of the successive samples of each harmonic.

17.2.6.8 Estimation of Noise Component of HNM

For unvoiced speech the excitation is a noise-like signal across the entire speech bandwidth. For voiced speech the excitation is a noise-like signal above some variable maximum harmonic frequency.

The main effect of the background noises on the estimate of the excitation of the LP model is an increase in its variance. In fact, perceptually good results can be obtained by replacing the noise part of the excitation to the LP model with a Gaussian noise with the appropriate variance estimated as the difference between the variance of the noisy signal and that of the noise. Finally the synthetic HNM of excitation signal is obtained as

$$\hat{E}(f,m) = E_H(f,m) + E_V(f,m) + E_N(f,m) \tag{17.48}$$

where $E_{\mathrm{H}}(f,m)$ is the harmonic part of the excitation, $E_{\mathrm{V}}(f,t)$ is the noise part of the excitation, $E_{\mathrm{N}}(f,\ m)$ is the noise disturbance contaminating the excitation.

17.2.6.9 Kalman Smoothing of Trajectories of Formants and Harmonics

The Kalman filter equations for de-noising or smoothing of all the parameters of speech are essentially the same; for this reason we describe the Kalman smoothing of formant tracks. The formant trajectory is modelled by an AR process as

$$\hat{F}_k(m) = \sum_{i=1}^{P} c_{ki}\hat{F}_k(m-i) + e_k(m) \tag{17.49}$$

where c_{ki} are the coefficients of a low-order (3 to 5) AR model of the k^{th} formant track and $e_k(m) = N(0, Q_k)$ is a zero mean Gaussian random process. The variance of $e_k(m)$, Q_k is estimated from the previous estimates of e_k. The algorithm for Kalman filter adapted for formant track estimation is as follows.

Time updates (prediction) equations

$$\hat{\boldsymbol{F}}_k(m|m-1) = \boldsymbol{C}\hat{\boldsymbol{F}}_k(m-1) \tag{17.50}$$

$$\boldsymbol{P}(m|m-1) = \boldsymbol{P}(m-1) + \boldsymbol{Q} \tag{17.51}$$

Measurement updates (estimation) equations

$$\boldsymbol{K}(m) = \boldsymbol{P}(m|m-1)\ \ (\boldsymbol{P}(m|m-1) + \boldsymbol{R})^{-1} \tag{17.52}$$

$$\hat{\boldsymbol{F}}_k(m) = \hat{\boldsymbol{F}}_k(m|m-1) + \boldsymbol{K}(m)\ \left(\boldsymbol{p}_k(m) - \hat{\boldsymbol{F}}_k(m|m-1)\right) \tag{17.53}$$

$$\boldsymbol{P}(m) = (\boldsymbol{I} - \boldsymbol{K}(m))\boldsymbol{P}(m|m-1) \tag{17.54}$$

where $\hat{\boldsymbol{F}}_k(m|m-1)$ denotes a prediction of $\boldsymbol{F}_k(m)$ from estimates of the formant track up to time $m-1$, $\boldsymbol{P}(m)$ is the formant estimation error covariance matrix, $\boldsymbol{P}(m|m-1)$ is the formant prediction error covariance matrix, $\boldsymbol{K}(m)$ is the Kalman filter gain, $\boldsymbol{R}$ is the measurement noise covariance matrix, estimated from the variance of the differences between the noisy formant observation and estimated tracks. The covariance matrix $\boldsymbol{Q}$ of the process noise is obtained from the prediction error of formant tracks.

Kalman theory assumes the signal and noise can be described by linear systems with random Gaussian excitation. Kalman filter is unable to deal with relatively sharp changes in the signal process, for example when speech changes from a voiced to a non-voiced segment. However, state-dependent Kalman filters can be used to solve this problem. For example a two-state voiced/unvoiced classification of speech can be used to employ two separate sets of Kalman filters: one set of Kalman filters for voiced speech and another set for unvoiced speech. In HMM-based speech models in each state of HMM the signal trajectory can be modelled as a Kalman filter.

The databases used for the evaluation of the performance of the speech enhancement systems are a subset of five male speakers and five female speakers from the *Wall Street Journal* (WSJ). For each speaker, there are over 120 sentences. The speech signal is down-sampled to 10 kHz from an original sampling rate of 16 kHz. The speech signal is segmented into overlapping frames of length 250 samples (25 ms) with an overlap of 150 samples (15 ms) between successive frames. Each speech frame is windowed with a Hann window.

Figure 17.13 shows the significant improvement in the harmonicity measure resulting from a FTLP-HNM model. Figure 17.14 illustrates the results of perceptual evaluation of speech quality (PESQ) of noisy speech and speech restored with MMSE and FTLP-HNM methods. It is evident that the LP-HNM method achieves improved results. Figure 17.15 shows the improvement in ISD measure compared with the MMSE system. It is evident that the new speech processing system achieves a better ISD score.

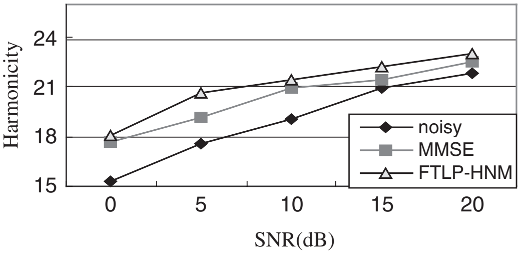


Figure 17.13 Comparison of harmonicity of MMSE and FTLP-HNM systems on train noisy speech at different SNRs.

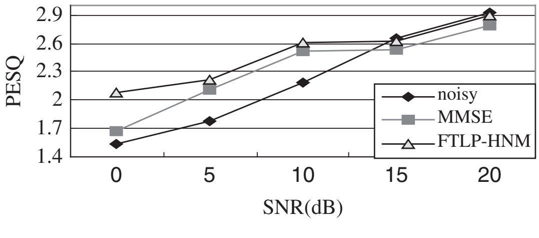


Figure 17.14 Performance of MMSE and FTLP-HNM on train noisy speech at different SNRs.

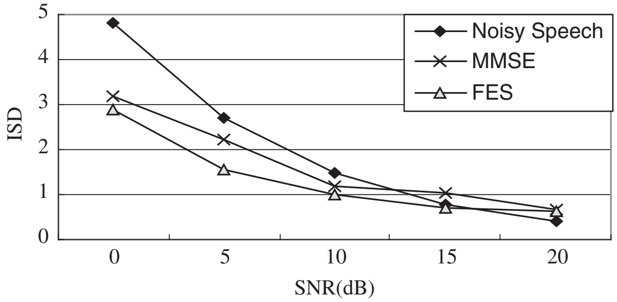


Figure 17.15 Comparison of ISD of noisy speech in train noise pre-cleaned with MMSE and improved with formant-based enhancement system (FES) at SNR = 0, 5, 10, 15, 20 dB.

17.3 Speech Bandwidth Extension–Spectral Extrapolation

Conventional telephony speech is limited to a narrow bandwidth of less than 4 KHz; normally 300–3400 Hz. In contrast, wideband broadcast quality speech can be as wide as 20–20 000 Hz. Although most of the energy and the information content of speech are contained within the telephony bandwidth of 300–3400 Hz, there exists significant energy in other frequencies that conveys quality and sensation. In particular the upper frequency bands above 3400 Hz convey significant information regarding the identity of the consonants and helps to differentiate for example an '*s*' from an '*f*'.

The aim of bandwidth extension methods is to reconstruct the missing upper (and lower) band spectrum of speech signals using an extrapolation of the available spectrum in order to regain the sensation of wide bandwidth and higher quality speech. Two major remaining research challenges in development of bandwidth extension systems are (1) the recovery of information for the class of speech signals where valuable information resides in the (missing) upper bands rather than in the available lower bands (e.g. fricatives) and (2) language dependency of current systems i.e. the lack of language portability.

This section describes an example of a bandwidth extension system developed by Zavarehei and Vaseghi. Most bandwidth extension techniques, such as that shown in Figure 17.16, use codebook mapping methods for extraction of the missing bands from the available information. The parameters of the missing spectral envelope and excitation in the higher bands are obtained from codebooks trained on joint feature vectors of narrowband and wideband speech. The spectral envelope representation for codebook mapping is often based on the line spectral frequency (LSF) parameters derived from a linear prediction model of speech.

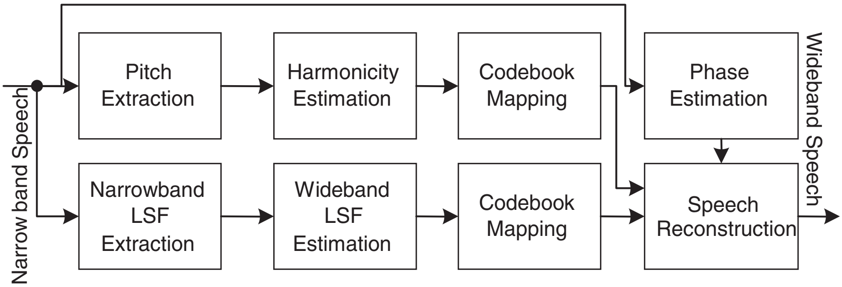


Figure 17.16 Block diagram of a bandwidth extension system based on LP-HNM model of speech. The parameters of the LP-HNM for missing speech are extracted from codebooks trained on joint feature vectors of narrowband and wideband speech.

An effective method for extrapolation of excitation is based on a harmonic plus noise (HNM) model of speech excitation. For voiced speech the fundamental frequency and harmonics of excitation are tracked and extrapolated. An essential aspect of the reproduction of harmonics is the idea of the extrapolation of the 'quality' of harmonics of speech. For this purpose a measure of harmonicity of speech excitation is defined. Harmonicity of each harmonic is a measure of the harmonic to fricative (aspiration noise) energy ratio at and around that harmonic. This measure is used to characterise both the harmonic structure of voiced speech and the non-harmonic structure of unvoiced speech. This approach has an added advantage as it circumvents the need for hard decisions regarding classification of speech into voiced/unvoiced parts. Figure 17.17 shows an example of a speech segment synthesised using a harmonic plus noise model where the proportion of the harmonic and noise components at each harmonic frequency is obtained from the harmonicity of speech.

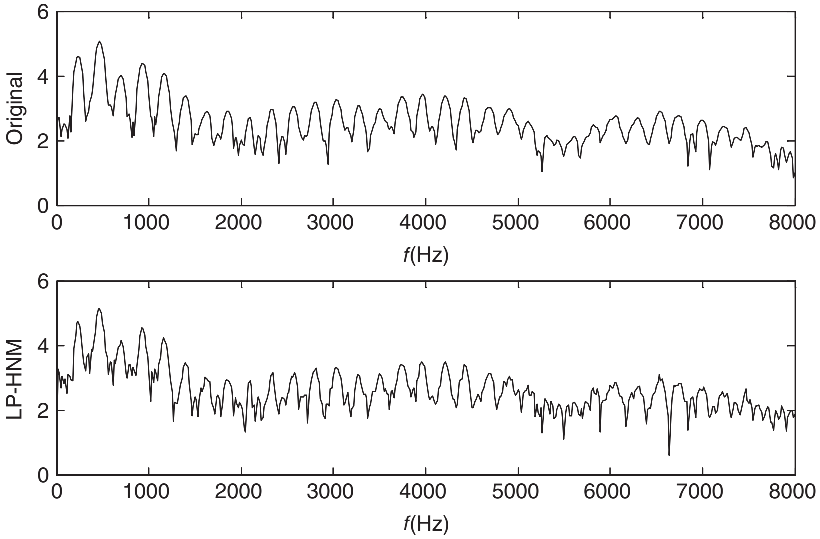


Figure 17.17 Harmonicity-based reconstruction of speech segment: original (top) and LP-HNM reconstructed (bottom).

The estimates of the spectral envelope are then combined with an estimate of the excitation signal and converted to time domain to yield the wideband output speech signal.

17.3.1 LP-HNM Model of Speech

The speech model used here for bandwidth extension, shown in Figure 17.16, is a linear prediction (LP) model of the spectral envelope and a harmonic noise model (HNM) of speech excitation. The parameters of the LP-HNM model are the fundamental frequency F_0, the adjusted harmonic frequencies F_k, the harmonic amplitudes A_k, the harmonicity values V_k, the LSF vector representation of the LP model coefficients $\boldsymbol{Q}$ and the LP model gain C.

The use of the degree of harmonicity (i.e. the ratio of the harmonic to aspiration noise-like energy around each harmonic frequency) for this application, results in a soft decision during codebook mapping rather than a hard decision. Moreover, during temporal fluctuations of the spectrogram, the harmonics gradually gain or lose strength which is modelled using the harmonicity degree.

During the reconstruction of the high-frequency part of the spectrum, the excitation amplitudes in each sub-band, at around the harmonics $E(kf_0)$, is assumed to be 1 which is reasonable assuming that the order of the LP model is high enough to have whitened the excitation. The gain of the frame is modelled using the LP gain value. The excitation reconstructed from this model is multiplied by the LP spectrum and the phase component to form the speech spectrum of each frame.

For the excitation codebook to have a fixed size, the harmonicity of a maximum of 32 bands is calculated for each frame. This is a design parameter that may vary in different systems. Sub-bands above this range are assumed to be entirely unvoiced. Experiments indicate that the efficiency of the system seems to saturate for a larger number of harmonics. A codebook of harmonicity feature vectors is trained on wideband speech signals from a training database such as the Wall Street Journal (WSJ) speech database using the K-means algorithm. The distortion measure is the Euclidean distance measure which is applied only to the lower 4 kHz of the signal. The upper 4 kHz harmonicity values act as a shadow codebook for retrieving the harmonicity of the upper sub-bands.

The initial fundamental frequency of excitation F_0 can be obtained using a correlation based method described in 17.2.6 and in (Zavarehie PhD 2008), which may be refined by pick-peaking and finding the harmonic frequencies and then adjusting the fundamental frequency according to harmonic frequencies. Furthermore, at each stage a number of pitch candidates corresponding to several peak values (local peaks and 'global' peaks) of the correlation or power-spectrum curves are selected and the Viterbi algorithm is used to obtain the best candidate and track the fundamental frequency.

17.3.2 Extrapolation of Spectral Envelope of LP Model

This section describes a method for extrapolation of the spectral envelope using codebooks trained on narrowband and wideband speech. For extrapolation of the spectral envelope of speech, the LP model coefficients are usually converted to line spectral frequency (LSF) coefficients as these have good quantisation and interpolation/extrapolation qualities. A 12th order speaker-independent LSF codebook is trained based on narrowband speech signals derived from a database such as the Wall Street Journal (WSJ) speech database down-sampled to 8 kHz sampling rate (4 kHz bandwidth). A shadow codebook, of 24th order LSF, for wideband speech sampled at 16 kHz rate is trained in conjunction with the 8 kHz sampled codebook. (Note that the bandwidth of the wideband signal may be chosen higher depending on the application.) The distortion measure used both in the training stage and in the estimation stage is the mean square spectral distortion (error) of the corresponding LP spectra:

$$D(X_{LSF}, Y_{LSF}) = \int_{-0.5}^{0.5} |X_{LP}(f) - Y_{LP}(f)|^2 \, df \tag{17.55}$$

where X_{LSF} and Y_{LSF} are the set of narrowband LSF vectors and $X_{\mathrm{LP}}(f)$ and $Y_{\mathrm{LP}}(f)$ are the corresponding LP spectra.

Estimation of the gain of the LP model, G, is crucial in bandwidth extension. Two different approaches for gain estimation were implemented: (1) estimation of the gain using energy normalisation and (2) codebook mapping.

In the first approach the gain of the wideband LP model is calculated to result in the same amount of energy in the low-band portion as that of the narrowband speech signal. While this approach works well in the voiced segments, experiments show that it does not result in good estimates for the frames where most of the energy is concentrated in the high frequency (e.g. fricatives).

In the second approach, the ratio of the gain of the narrowband LP model of each frame, $G_{\mathrm{narrowband}}$, divided by that of the wideband signal for the same frame, G_{wideband}, is calculated:

$$R_G = G_{narrowband} / G_{wideband} \tag{17.56}$$

The gain ratio R_G is calculated for each frame and a shadow codebook is trained in conjunction with the LSF codebook, similar to the shadow codebook for the wideband LSF.

During the estimation stage, the narrowband LSF values are calculated for each frame and the closest (measured in terms of mean squared error spectral distortion) codeword is chosen from the narrowband LSF codebook. The corresponding wideband codeword and gain ratio are obtained from the shadow codebooks. The narrowband LP model gain is divided by the gain ratio and the result is used as the wideband LP model gain.

The LSF, gain and harmonicity codebooks are trained using speech signals obtained from the WSJ speech database, spoken by several speakers. The WSJ speech is originally sampled at 16 kHz and has a bandwidth of 8 kHz. Speech is segmented to frames of 25 ms duration with a frame overlap of 15 ms, i.e. a frame rate of 10 ms. To produce narrowband speech, wideband speech is filtered to a bandwidth of 4 kHz and down-sampled to a sampling rate of 8 kHz.

For the purpose of system evaluation, 40 test sentences are randomly chosen from the WSJ speech database, the test sentences were not among those used for training the codebooks. The mean log spectral distance (LSD) between the wideband and bandwidth-extended signals are calculated and averaged over all frames and sentences for each modification. The average LSD is calculated using the fast Fourier transform (FFT) of the frames (FFT-LSD) and LP spectrum of the frames (LP-LSD).

17.3.2.1 Phase Estimation

There are several different methods for coding and estimation/prediction of the phase of the missing spectra such as phase codebooks and phase predictors. As most of the signal in the higher bands of the wideband speech is not harmonically structured, the issue of phase estimation is not extensively explored in the literature of bandwidth extension. In this example work the phase of the upper band is estimated from the lower band so that the unwrapped phase of each frame is linear. Some random phase, proportional to the inverse of the harmonicity of each sub-band is then added to the phase to account for the non-harmonic random phase. In experimental evaluations, absolutely no perceptible difference is audible when the upper band phase of a wideband signal is replaced by its predicted value.

17.3.2.2 Codebook Mapping of the Gain

The estimation of the LP gain of the wideband signal is crucial in bandwidth extension of narrowband signals. The use of the codebook mapping method for estimation of the LP gain is compared with the energy normalisation method. It is observed that using energy normalisation results in suppression of the signals, especially during fricative segments where most of the energy of the signal is concentrated in the higher bands of the signal. Estimation of the LP gain value through codebook mapping results in superior quality of the wideband signals. A comparison of the LSD values of these two methods is presented in Table 17.1 which shows that codebook mapping results in lower averaged LSD distances in every case.

Table 17.1 Average log spectral distance of the wideband signal for energy normalization and codebook mapping of the gain factor.

Distortion measure (dB)	Energy normalisation	Codebook mapping
Overall FFT-LSD	5.83	5.10
High band FFT-LSD	8.25	7.17
Overall LP-LSD	4.96	4.27
High band LP-LSD	6.83	6.09

17.3.3 Extrapolation of Spectrum of Excitation of LP Model

The use of the harmonicity model for reconstruction of the excitation of the speech signal in missing frequency bands is compared with the alternative method of using a band-pass envelope modulated Gaussian noise (BP-MGN). While the rest of the system is similar for evaluations carried out here, only the excitation is estimated using the two different methods. It is observed that both methods result in reasonably good quality output speech. However, from the study of the spectrograms of the wideband signals produced by these systems it was observed that the extended bands which had more harmonically structured patterns were better reconstructed using the harmonicity model. Table 17.2 summarises the averaged LSD values calculated for these two cases. While the difference linear prediction based LSDs are large, those of the FFT-LSD seem to be higher. This is due to the more detailed modelling of the harmonics of the higher bands in the proposed method.

Table 17.2 Average log spectral distance of the wideband signal for the band-pass envelope modulated Gaussian noise (BP-MGN) model and the harmonicity model.

Distortion measure (dB)	BP-MGN model	Harmonicity model
Overall FFT-LSD	5.85	5.10
High band FFT-LSD	8.79	7.17
Overall LP-LSD	4.51	4.27
High band LP-LSD	6.63	6.09

17.3.3.1 Sensitivity to Pitch

To accurately estimate the harmonicity levels, it is crucial that reasonably accurate estimates of the fundamental frequency are extracted from the narrowband speech signal. Inaccurate fundamental frequency estimates normally do not happen in harmonically well-structured speech frames, which are also more likely to have higher frequency harmonics. It is observed that inaccurate fundamental frequencies during

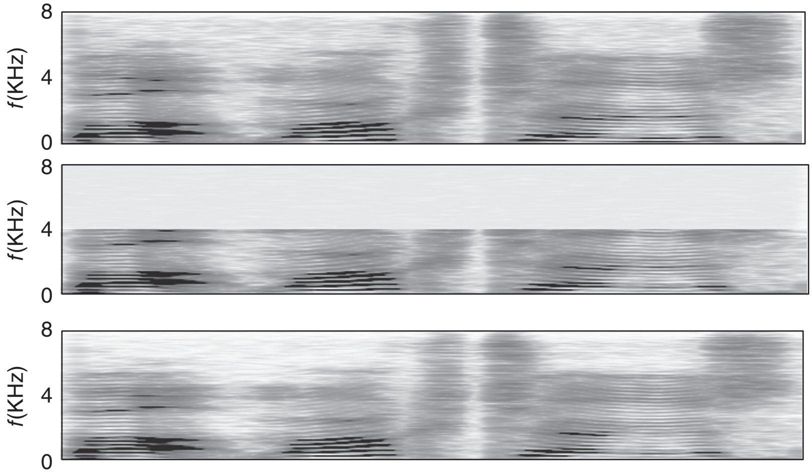


Figure 17.18 Spectrograms of (top) wideband signal, (middle) narrowband signal and (bottom) bandwidth extended signal.

harmonically structured (voiced) frames results in inaccurate extrapolation of harmonicity. However, this is not the case during unvoiced frames. During unvoiced frames the excitation signal is normally reconstructed using only noise as, even if a fundamental frequency is assigned to it, the harmonicity values calculated for sub-bands will be very low which will result in domination of the noise component.

Figure 17.18 shows an example of the original wideband speech with a bandwidth of 8 kHz, the filtered speech with a bandwidth of 4 kHz and the restored extended bandwidth speech. LP gain and harmonicity values are estimated using the proposed method of codebook mapping. The figure clearly shows both the harmonic and noise structures in the upper band have been well restored.

17.4 Interpolation of Lost Speech Segments–Packet Loss Concealment

Interpolation of lost speech segments is an essential requirement in a number of speech applications for the reconstruction of speech segments that are missing or lost to noise or dropouts such as for packet loss concealment (PLC) in speech communication over mobile phones or voice over IP (VoIP), restoration of archived speech recordings and for general purpose speech interpolation.

Algorithms specifically designed for speech gap restoration can be categorised into two classes:

(1) Predictive or extrapolative methods, where only the past samples are available.
(2) Estimative or interpolative methods, where some future samples are also available.

In general, speech interpolation methods utilise signal models that capture the correlations of speech parameters on both sides of the missing speech segment. The speech signal gap either can be interpolated in the time domain or one can convert speech in its spectrogram and then interpolate the missing spectral-time components. The latter method, which utilises the spectral temporal structure, is the focus of this section.

Autoregressive (AR) models are of particular interest in restoration of lost samples of speech. Janssen *et al*. applied AR models for estimation of non-recoverable errors in compact disk systems. Least square error AR (LSAR) and maximum a posterior (MAP) AR are among the most common criteria used for estimation of the gaps. Vaseghi and Rayner proposed a pitch-based AR interpolator to take into account the long-term correlation structure of the speech and near periodic signals. Kauppinen and Roth proposed a method that replaces the missing portion of the excitation with zeros which effectively uses the LP model for estimation of the missing samples from past samples only. Estimation of the excitation of the AR source filter is crucial in AR-based interpolators. Esquef *et al*. proposed a time-reversed excitation substitution algorithm with a multi-rate post-processing module for audio gap restoration.

Lindblom and Hedelin proposed methods for extrapolation (prediction) of lost speech packets using linear prediction; they used LP analysis for modelling the spectral envelope of the signal and sinusoidal models for modelling the excitation. They also considered the interpolation problem. Wang and Gibson (2000) investigated the robustness of the inter-frame and intra-frame line spectral frequency (LSF) coders to packet loss. .

Another strategy that has proved to be popular for PLC is waveform substitution. The International Telecommunications Union (ITU) has standardised a waveform substitution algorithm for extrapolation of speech signals up to 60 ms after first frame loss. Waveform substitution techniques search the past few received frames to find appropriate substitutions for the lost packet. This is an improvement on repeating the exact previous frame which can produce annoying artefacts. Goodman et al. reported a predictive waveform substitution technique for application in PLC. Valenzuela and Animalu proposed a pitch synchronous overlap-add interpolation method using a similar technique.

In other works combinations of the AR-based algorithms and waveform substitution are used (Gündüzhan *et al*., 2001) and (Elsabrouty *et al*., 2003). Wang and Gibson (2001) proposed the use of linearly interpolated LSFs throughout the gap and optimised selection of the coded excitation from previous frames for restoration of gaps in a CELP coded speech signal.

Sinusoidal models, and their extended version HNM, which have found extensive applications in text-to-speech (TTS) synthesis (Stylianou, 2001), coding (Griffin & Lim, 1988 and Kondoz, 1999), and many other speech processing applications including bandwidth extension (Vaseghi & Zavarehei). Rødbro *et al.* (2003) proposed a linear interpolation technique for estimation of the sinusoidal model parameters of missing frames. The time-domain signal, then, is synthesised using a combination of symmetric and asymmetric windows and an overlap-add technique.

In this section an interpolation method is described based on a LP-HNM model of speech where the spectral envelope of speech is modelled using a LSF representation of a linear prediction (LP) model and the excitation is modelled with a harmonic plus noise (HNM), whose parameters are the harmonic frequencies, amplitudes and harmonicities (voicing levels). These parameters are interpolated throughout the gap resulting in natural sounding synthesised speech. The advantage of using LP-HNMs for modelling the spectral envelope and excitation is that the time-varying contours of the formants and the harmonic energies across the signal gap are modelled and tracked.

Furthermore, a codebook-mapping technique is used as a post-processing module for fitting the interpolated parameters to a prior speech model which is trained beforehand. Interpolation of LSF values of speech may sometimes result in estimation of some unusually sharp poles, giving rise to tonal artefacts. The conventional method of mitigating such effects is bandwidth expansion through damping of the poles of the LP model at the cost of smearing the signal in the frequency domain (broadening the poles' bandwidth). Application of the codebook-mapping technique to interpolated parameters of the LP-HNM of the signal mitigates the effects of unwanted tonal artefacts resulting from the interpolation of LSF parameters without an undesirable broadening of the poles' bandwidths. Codebook-mapping results in significant improvement in the performance of the algorithm in restoration of longer gaps (e.g. > 50 ms).

The LP-HNM based speech interpolation method combines the advantages of the different methods described above and has the following distinct features:

(1) The speech structure is modelled by a time-varying LP-HNM model; i.e. a combination of an LP model of the spectral envelope and a HNM of the excitation.
(2) Interpolation is performed in spectral-time domain, where the non-stationary dynamics of the prominent contours of energies of the spectral envelope (LSF) and the excitation tracks (pitch, amplitudes, phase and harmonicities) of speech on both sides of the missing signal gap are modelled and interpolated.
(3) The temporal dynamics of the LP-HNM parameter tracks are modelled by a combination of linear and AR models.
(4) A pre-trained post-processing codebook is employed to mitigate artefacts and enhance the interpolation process; in this way prior statistical model information is brought into the interpolation process.

Figure 17.19 shows the interpolation method where speech interpolation is transformed into interpolation of the LP-HNM frequency-time tracks. The interpolation of each LP-HNM track is achieved using two methods: (1) a simple linear interpolation, (2) a combination of linear and autoregressive interpolation methods.

Assume that T_G consecutive frames of speech are missing where each speech frame has W samples including S new (non-overlapping) samples; and let T_A and T_B be the number of available speech frames after and before the speech gap respectively. Our goal is to estimate the LP-HNM parameters of T_G missing frames using the T_B frames before and T_A frames after the gap as shown in Figure 17.20. It is assumed that the interpolation delay is less than the maximum acceptable system delay.

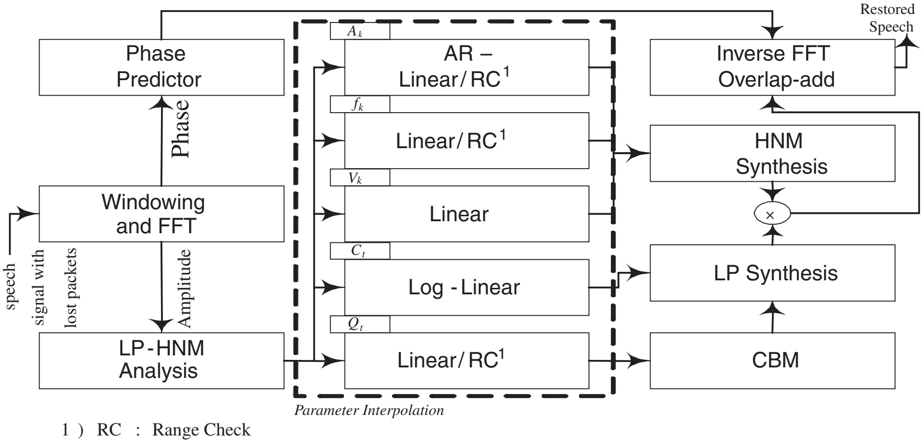


Figure 17.19 LP-HNM + CBM interpolation systems. LP-HNM: linear prediction harmonic noise model, CBM: codebook mapping.

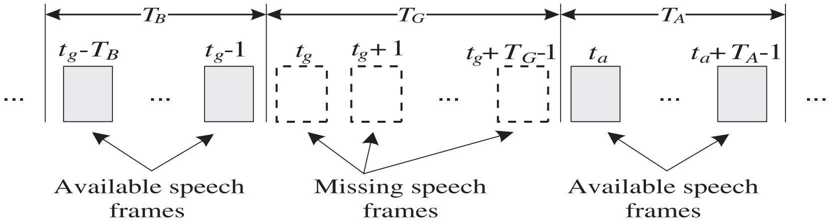


Figure 17.20 Illustration of speech frame interpolation where T_G frames of speech signal are missing.

Assuming $T_B = T_A = 1$, i.e. only one speech frame is available before and after the gap, the LP-HNM parameters can be linearly interpolated. We define the general linear interpolating function as

$$I_L(x_1, x_2, T_G, t) = \frac{tx_2 + (T_G - t + 1)x_1}{T_G + 1} \qquad 1 \leq t \leq T_G \tag{17.57}$$

where x_1, x_2 are the known values at the two ends of the gap. Each LP-HNM parameter track composed of the harmonic amplitudes A_{kt}, the harmonicity V_{kt}, the harmonic frequencies f_{kt}, the LSF coefficients Q_{it} and the LP model gain C_t, can be linearly interpolated using Equation (17.57). Note the subscripts k, i and t represent the harmonic, LSF and frame indices respectively.

The linear interpolation method joins the LP-HNM parameters of speech across the gap with a straight line. As expected the quality of the interpolated speech is sensitive to estimation error of the excitation harmonic amplitudes and LSF values. Furthermore experiments show that classical high-order polynomial interpolators result in artefacts in the output.

The response of a stable AR model with non-zero initial conditions to a zero excitation decays with time towards zero. The proposed interpolation method exploits this fact in order to obtain an estimate of the parameter sequence which has a smooth transition at each side of the gap and is modelled by the mean values of the LP-HNM parameters in the middle.

Assume the values of the time series x_t are missing from the time instance t_g to $t_a - 1$. One solution would be the LSAR interpolator which incorporates information from both sides of the gap simultaneously. However, LSAR assumes that the signals on both sides of the gap are from a stationary process. Furthermore, a large number of samples are required for a reliable estimate of the LSAR models of the time series before and after the gap.

In the method shown in Figure 17.21 two low-order AR models are used for the estimation of the zero-mean trend of the time-series parameter from each side of the gap. The predicted values from each side are overlap-added. The mean value of the time series is estimated by linear interpolation of the mean values of both sides of the gap. Let the number of available frames at each side of the gap $T_A = T_B \geq 3$. The missing parameter values are estimated as

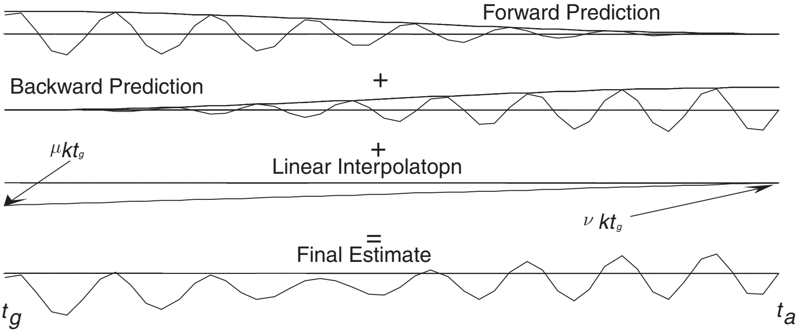


Figure 17.21 The mean-subtracted time-series is linearly predicted from both sides, weighted-averaged and added to the linearly interpolated mean.

$$X_t^{(\mathrm{AR})} = W_t \sum_{i=1}^{P} a_{t_g}^{(i)} \left(x_{t-i} - \mu_{t_g}^{(x)}\right) + (1 - W_t) \sum_{i=1}^{P} b_{t_a}^{(i)} \left(x_{t+i} - v_{t_a}^{(x)}\right) + I_L\left(\mu_{t_g}^{(x)}, v_{t_a}^{(x)}, T_G, t - t_g + 1\right) \tag{17.58}$$

where P is the LP order of the AR model and

$$\mu_{t_g}^{(x)} = \frac{1}{T_B} \sum_{i=1}^{T_B} x_{t_g - i} \text{ and } v_{t_a}^{(x)} = \frac{1}{T_A} \sum_{i=1}^{T_A} x_{t_a + i - 1} \tag{17.59}$$

are the means and $a_{t_g}^{(i)}$ and $b_{t_a}^{(i)}$ are the i^{th} LP coefficients of the series $\left[x_{t_g - T_B}, \ldots, x_{t_g - 1}\right] - \mu_{t_g}^{(x)}$ and $\left[x_{t_a + T_A}, \ldots, x_{t_a + 1}\right] - v_{t_a}^{(x)}$, respectively. The weights, W_t, are chosen from half of a Hanning window of length $2T_G$. Figure 17.22 shows an example of the AR interpolation used for interpolation of the fifth harmonic of a sample signal. The gap, T_G, is rather long (i.e. equal to 70 ms).

17.4.1 Phase Prediction

Short-time phase plays an important role in the perceived quality and naturalness of the HNM-synthesised speech. In order to obtain an acceptable quality of reconstructed speech, the phase estimation method must be based on a model that exploits the continuity of the harmonic parts of speech and maintains the randomness of the non-harmonics. In this section a method is introduced for one-frame forward

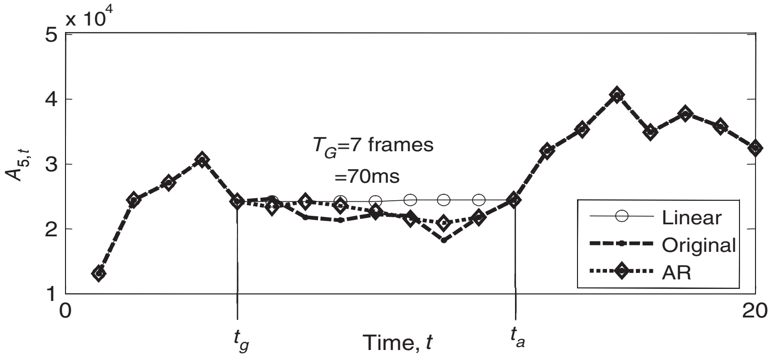


Figure 17.22 AR interpolation of the fifth harmonic of a sample signal.

prediction of speech phase, based on the HNM model of speech. In order to maintain inter-frame and intra-frame continuity of the reconstructed signal, the phase is estimated so that:

(1) speech harmonics across successive frames have phase continuity (inter-frame continuity, Figure 17.23 (left);
(2) the adjacent frequency bins around each harmonic are in phase halfway through the frame (intra-frame continuity, Figure 17.23 (right); and
(3) the amount of randomness added to the phase of each frequency channel increases with its frequency and its distance from the nearest harmonic centre (unvoiced sub-bands, Figure 17.24).

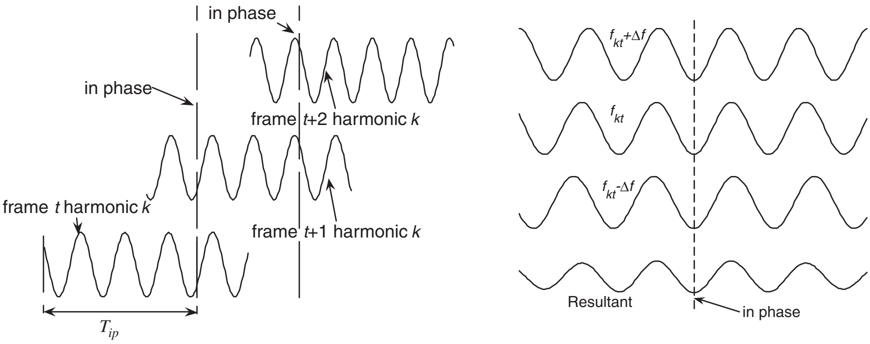


Figure 17.23 Harmonics of successive frames are in phase (left); adjacent bins of harmonics are in phase with them half-way through the frame (right).

Condition 1 guarantees the continuity between successive frames. Condition 2 determines the slope of the linear phase around each harmonic. This ensures positive reinforcement of the energy of signal components around each harmonic. Condition 3 is useful in reconstruction of unvoiced sub-bands and helps to avoid tonality and unwanted artefacts. In order to satisfy the first condition, as shown in Figure 17.23 (left), the equation for phase at harmonics is obtained as

$$\Phi(f_{kt}) = \Phi(f_{k,t-1}) + \frac{2\pi}{F_s}\left[T_{\text{ip}}\left(f_{k,t-1} - f_{k,t}\right) + sf_{k,t}\right] \tag{17.60}$$

where $\Phi(f_{kt})$ is the phase of the k^{th} harmonic at frame t (i.e. with frequency f_{kt}), T_{ip} is the 'in-phase' sample index that is where harmonic f_{kt} and $f_{k,t-1}$ are in phase, S is the shift size and F_s is the sampling frequency.

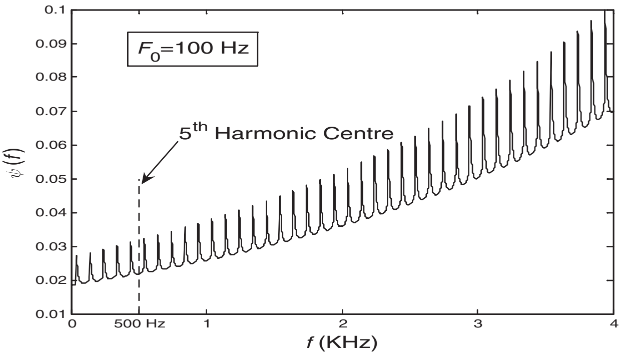


Figure 17.24 Variations of the amount of noise added to phase across frequency.

Note that the subscript of the harmonic frequency has been extended to include the frame index as well. This subscript might be used for other LP-HNM parameters as well wherever necessary. The in-phase sample is chosen to be halfway through the overlap as shown by vertical lines in Figure 17.23 (left), i.e.

$$T_{\text{ip}} = (W + S)/2 \tag{17.61}$$

where W is the window size. In order to obtain positive energy reinforcement at the middle of each window, the frequency channels in the vicinity of each harmonic should be in phase at that point (Figure 17.23 (right)). Furthermore a level of randomness needs to be added to the phase for unvoiced (noise) synthesis, i.e. for each harmonic sub-band of an arbitrary frame:

$$\Phi(f) = \Phi(f_k) + (f - f_k)a + \frac{\Phi_R(f)}{V_k}\psi(f) \text{ for } 0 < |f - f_{\text{k}}| < F_0/2 \tag{17.62}$$

$$a = -\pi W/F_S \tag{17.63}$$

where a is the slope of the phase, $\Phi_R(f)$ is a random variable uniformly distributed in the range $[-\pi,\ \pi]$, $\psi(f)$ is a weighting factor that increases with the frequency and distance from the centre of the harmonic. The latter determines the level of randomness of the phase. We propose the use of the following function for this purpose:

$$\psi(f) = \sum_{k=1}^{N}\left[\left(\frac{1 - h(f - f_k)}{h(f - f_k)}\right)0.002 + 0.05\right]\exp\left(\frac{f - 3000}{3000}\right) \tag{17.64}$$

where $h(f)$ is a Hamming window in the range $[-F_0/2,\ F_0/2]$. An example of $\psi(f)$ is shown in Figure 17.24. Note that the local minimums correspond to harmonic frequency channels. This figure is further shaped according to the harmonicity of each sub-band.

17.4.2 Codebook Mapping

Codebook-mapping (CBM) is a heuristic technique normally used for partial estimation of a set of parameters, e.g. estimation of upper band parameters based on lower band for bandwidth extension, or correction of over-suppressed harmonics in a noise reduction system. Codebook mapping forces a model upon the parameters through use of pre-trained codebooks.

In estimation of LSF parameters, using linear interpolators, we note that the resulting spectral envelope may have sharp peaks or sound unnatural. These artifacts can be even more annoying than the original packet loss introduced. One technique that can be particularly useful is damping the poles of the LP model, perhaps proportional to the distance from the two ends of the gap. This would mitigate the problem of perceiving sharp peaks in the spectrum at the cost of a de-shaped spectrum.

The codebook mapping technique can be used to fit the estimated values into a pre-trained speech model through the use of codebooks. A codebook is trained on LSF parameters of various speech utterances. The utterances were taken from the Wall Street Journal (WSJ) database of spoken language. Each interpolated LSF vector is then compared with the vectors in the codebook and the K nearest codewords, $[\overline{\mathbf{Q}}_{k_1}, \cdots, \overline{\mathbf{Q}}_{k_k}]$, are selected according to the Euclidian distance:

$$D_k = \|\mathbf{Q}^{(\mathrm{L})} - \overline{\mathbf{Q}}_k\| \tag{17.65}$$

where $\mathbf{Q}^{(\mathrm{L})}$ is a linearly-interpolated LSF vector, $\overline{\mathbf{Q}}_k$ is the k^{th} codeword of the LSF codebook, D_k is the Euclidian distance between the two and $k_1,\ k_2, \ldots, k_K$ are the indices of the nearest codewords to $\mathbf{Q}^{(\mathrm{L})}$. These codewords are weighted averaged where the weights are inversely related to their distances from the original LSF vector. The resulting vector replaces the interpolated LSF vector.

$$\mathbf{Q}^{(\mathrm{CBM})} = \left[\sum_{i=1}^{K} \frac{1}{D_{k_i}}\right]^{-1} \times \sum_{i=1}^{K} \frac{\overline{\mathbf{Q}}_{k_i}}{D_{k_i}} \tag{17.66}$$

where the superscript (CBM) shows the codebook-mapped estimate of the LSF vector. Three different versions of the proposed algorithm are evaluated and compared with some alternative methods in this section. Besides parametric LP-HNM interpolation with and without codebook mapping, a different method which interpolates the HNM parameters extracted from the speech spectrum itself (and not the excitation) is also evaluated.

The multirate gap restoration algorithm, introduced by Esquef and Biscainho (2006), is chosen for comparison purposes. This algorithm is composed of two modules: (1) a *core* module which uses an AR model for each side of the gap and estimates the signal using an estimated excitation signal, and (2) a *multirate* post-processing module, which further enhances the interpolated signal in two low-frequency sub-bands. In addition to the complete algorithm, the performance of the core method (i.e. without the multirate post-processing) is also evaluated and compared with the proposed algorithms.

Many PLC algorithms proposed in the literature are compared with the standard ITU-T G.711 PLC algorithm. Even though the G.711 PLC algorithm is based on a different set of assumptions than the proposed algorithm, its performance is evaluated compared with the proposed algorithm as a reference point.

A two-state Markov model is used to model the frame loss introduced in the speech signal. The probability of a 'lost' frame after a 'good' frame is p and that of a good frame after a bad frame is q. This model emphasises the burst errors that might occur in some applications.

17.4.2.1 Evaluation of LP-HNM Interpolation

After introducing the gaps in the signals, each signal is restored using different algorithms, e.g. ITU G.711 PLC algorithm (G.711), multirate gap restoration (Multirate), the core AR-based algorithm of Janssen et al and the proposed algorithms (HNM, LP-HNM, LP-HNM+CBM).

The performance of these algorithms is evaluated using perceptual evaluation of speech quality (PESQ) scores and log spectral distance (LSD) measure. The results are calculated and averaged for 100 sentences randomly selected from the WSJ database. The performances of different algorithms in restoration of the gaps generated by a two-state Markov model are illustrated in Table 17.3. Note that the probability of a 'bad' or lost frame after a 'good' or available frame is p and that of a good frame after a bad frame is q.

A set of five utterances are selected randomly from the WSJ database. Three different sets of packet loss patterns are generated, using the two-state Markov model described in the previous section, with a fixed loss rate of 40 % and different average gap lengths of 2, 5 and 7 frames. An experiment similar to

Table 17.3 PESQ and LSD performance of different algorithms for restoration of two-state Markov generated gaps for four different values of the parameters, *p, q* and the resulting loss rate and average gap length.

	PESQ				LSD			
q	0.85	0.7	0.5	0.4	0.85	0.7	0.5	0.4
p	0.1	0.2	0.3	0.6	0.1	0.2	0.3	0.6
Loss rate %	11	22	38	60	11	22	38	60
Av. gap length	1.18	1.43	2.00	2.50	1.18	1.43	2.00	2.50
HNM	3.15	2.73	2.43	2.12	0.52	0.72	0.85	1.06
LP-HNM	3.15	2.74	2.44	2.13	0.52	0.74	0.86	1.00
LP-HNM + CBM	3.14	2.74	2.48	2.21	0.52	0.70	0.81	0.96
G.711–A1	3.14	2.59	2.07	1.51	1.59	1.99	2.09	2.19
AR	3.00	2.60	2.25	1.77	0.42	0.56	0.64	0.80
Multirate	2.54	2.07	1.73	1.24	0.58	0.77	0.88	1.11
Distorted	2.76	2.01	1.18	0.44	–	–	–	–

ITU's Comparison Category Rating (CCR) is conducted. After introduction of the gaps, each signal is restored using the three proposed methods and the G.711 method. Ten listeners were asked to listen to the resulting signals, each played after its G.711 restored counterpart and compare the second utterance with the first one and rate it from −3 to 3 representing 'much worse' and 'much better' respectively. The results are summarised in Table 17.3.

Figure 17.25 shows the spectrograms of a part of a speech signal: its distorted (with missing samples) and restored versions. It is evident that the restored consonant in the middle of the sample (before 0.4 ms)

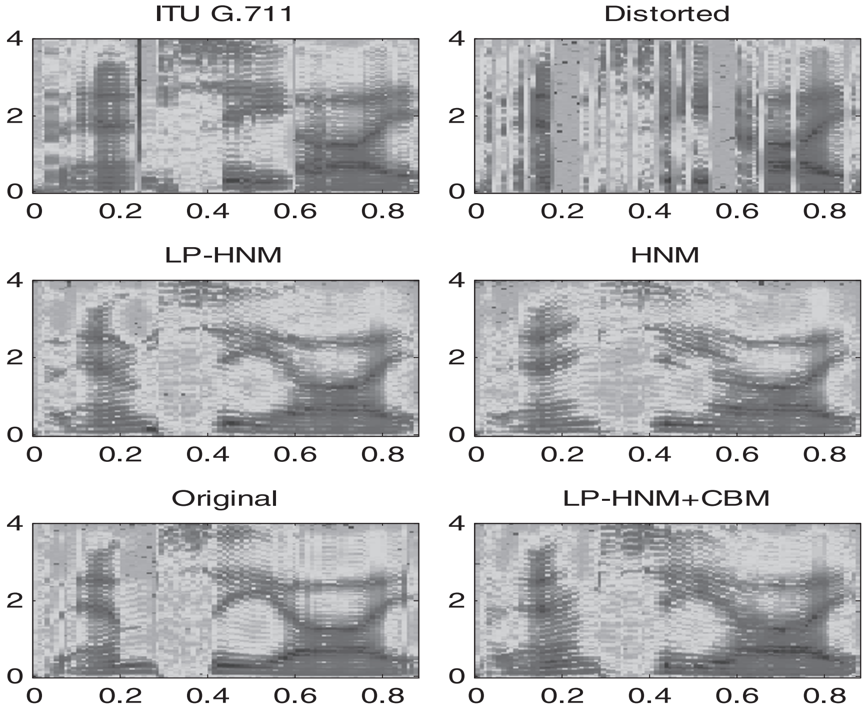


Figure 17.25 Spectrogram of a sample signal, with introduction of 40 % Bernoulli frame loss and the restored versions.

suffers from different artefacts in different methods. The upper bands of the speech signal, after restoration with the proposed algorithms, have a higher level of harmonicity compared with the interpolation method used in G.711. This is due to the more harmonic start of the consonant, available to the algorithms. A freezing effect can be seen throughout the restored gaps of the G.711 algorithm which is a known problem of this method. Furthermore, it is observed that the formant trajectories are best recovered using LP-HNM based algorithms.

It is rather difficult to evaluate and compare the performance of speech gap restoration algorithms. Not only is each designed for a particular application and uses specific resources available, but also they perform differently in reconstruction of different parts of speech signals. Through exhaustive experiments it is concluded that gap restoration algorithms, in general, are less successful in restoration of vowel–consonant and consonant–vowel transition and even less successful in restoration of vowel–consonant–vowel in which the restored quality is reduced to that of mumbled speech.

The objective results represented in the previous section shows that the LP-HNM method outperforms other algorithms discussed here in most cases. While a very similar output quality is gained in restoration of short gaps, the proposed algorithms are particularly powerful in restoration of longer gaps. The CBM technique results in a level of noise which is believed to be the result of the quantisation of the LSF vectors. While at shorter gap lengths this reduces the quality in comparison with some other methods, it makes the algorithm more robust to increases in the gap length particularly for gap lengths greater than 5 frames, as evident in Table 17.4.

Table 17.4 Comparative subjective results of proposed methods with a loss rate of 40 %.

Restoration method	HNM			LP-HNM			LP-HNM + CBM		
q	0.5	0.2	0.14	0.5	0.2	0.14	0.5	0.2	0.14
p	0.3	0.13	0.95	0.3	0.13	0.95	0.3	0.13	0.95
Av. gap length	2	5	7	2	5	7	2	5	7
Subjective score	1.64	1.12	0.61	2.36	1.88	0.73	1.68	1.60	1.29

Figure 17.25 shows the spectrogram of a sample signal, with introduction of 40 % Bernoulli frame loss and the restored versions. It is clearly evident that LP-HNM+BCM produces the better results.

17.5 Multi-Input Speech Enhancement Methods

In multiple input noise reduction systems several noisy input signals, picked up by an array of microphones, are filtered, time aligned and combined to reinforce the desired signals and reduce distortions due to noise, interfering speech, echo and room reverberations.

Multi-input speech enhancement systems include adaptive beam forming, adaptive noise cancellation, multi-input multi-output (MIMO) teleconferencing systems, stereophonic echo cancellation and in-car MIMO communication systems.

In a typical multi-input speech enhancement system, Figure 17.26, 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, 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

$$y_j(m) = \sum_{i=1}^{N} \sum_{k=0}^{P} h_{ij}(k) x_i(m-k) \qquad j = 1, \ldots, M \tag{17.67}$$

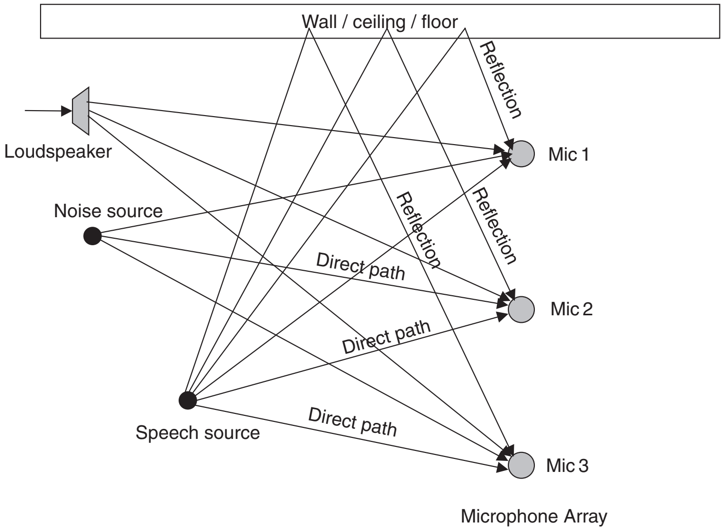


Figure 17.26 Illustration of different sounds and noise arriving at microphones, via direct line of sight paths and via reflections from walls/ceiling/floor.

where $h_{ij}(k)$ denotes the response of the channel from source i to microphone j modelled by a finite impulse response (FIR) linear filter. Note that for simplicity each source of signal, noise or interference is denoted with the same letter x and different index as $x_i(m)$; m is the discrete-time index.

In the simplest MIMO model, the response of an acoustic channel from sound source i to 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

$$h_{ij}(m) = \alpha_{ij}(m)\,\delta\left(m - \tau_{ij}(m)\right) \tag{17.68}$$

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 source i to microphone j needs to be expressed as

$$h_{ij}(m) = \sum_{k=1}^{L} \alpha_{ijk}(m)\,\delta\left(m - \tau_{ijk}(m)\right) \tag{17.69}$$

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.

Multi-input multi-output (MIMO) speech enhancement systems continue to be the subject of much 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 non-uniqueness of the solutions for room channel responses, when the noise sources are correlated. This is explained in the discussion on stereo acoustic echo cancellation in Chapter 15.
(3) The speed of convergence of adaptation of the filter 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.

The outputs of the MIMO systems can be processed in many different ways. One of the most common forms of MIMO systems is beam-forming microphone arrays for noise reduction, described next.

17.5.1 Beam-forming with Microphone Arrays

Microphone array beam-forming, Figure (17.27), is a noise reduction method in which an array of microphones and adaptive filters provide steerable directional reception of sound waves. The effect is that sounds arriving at microphones along the main beam of the array are constructively combined and reinforced whereas sounds (including disturbances such as noise, reverberation and echo) arriving from other directions are relatively attenuated. However, unwanted signals propagating together with the desired signal along the direction of the beam are not suppressed.

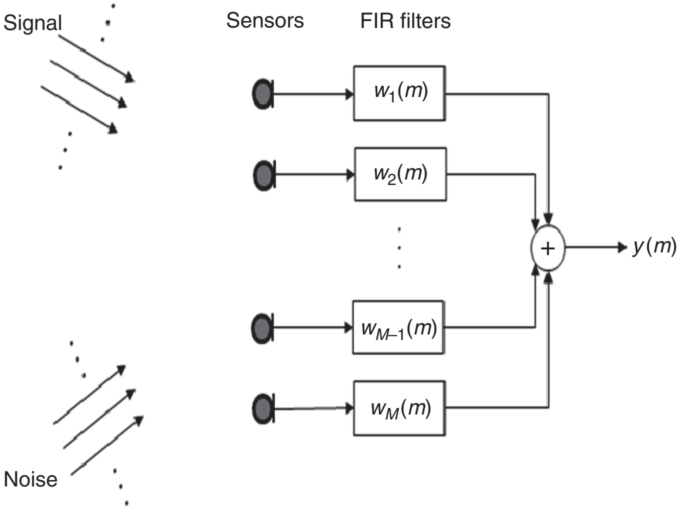


Figure 17.27 An array of sensor together with a filter and sum beam-former.

Note that it is assumed that the microphone is in the far field of the sound source and hence the sound waves reaching the microphone array can be considered as planar (as opposed to spherical) waves.

Beam-forming has applications in hands-free communication such as in-car communication, personal computer voice communication, teleconferencing and robust speech recognition. Beam-forming can also be combined with acoustic feedback cancellation.

As shown in Figure 17.27, beam-forming employs temporal-spatial filtering to steer the microphone array towards a desired direction. In its simplest form a beam-forming algorithm is a delay-sum device; the signal received at the k^{th} microphone is delayed by an amount τ_k such that all the signals coming from the desired (look out) direction are time-aligned and in-phase. The summation stage results in constructive reinforcement of the signals from the desired direction whereas the signals arriving from other directions are not time aligned, are out of phase and relatively attenuated after summation.

In this way it is possible to adaptively adjust the filters to selectively steer the array and pick up a sound wave from the direction of the source and/or where the sound energy is strongest and screen out noise, feedback and reflections of sounds from other directions.

Figure 17.28(a) shows the sound field of an omni-directional microphone that picks sound equally from all directions. The combined output of several omni-directional microphones placed in an array exhibits directional selectivity as shown in Figure 17.28(b).

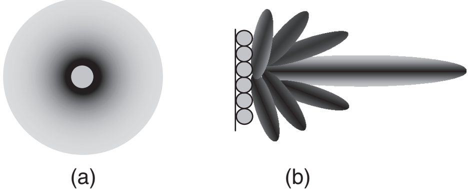


Figure 17.28 Illustration of the reception fields (aperture) of (a) an omni-directional microphone, (b) an array of microphones.

17.5.1.1 Spatial Configuration of Array and The Direction of Reception

Figure 17.29 shows a linear, uniformly-spaced, array of sensors with a distance of d between successive adjacent sensors. The sensors are shown in spherical co-ordinates where a unit length vector is represented as $\alpha(\theta, \phi)$ with zenith angle of $0 \leq \theta \leq \pi$ and azimuth angle of $0 \leq \phi \leq 2\pi$

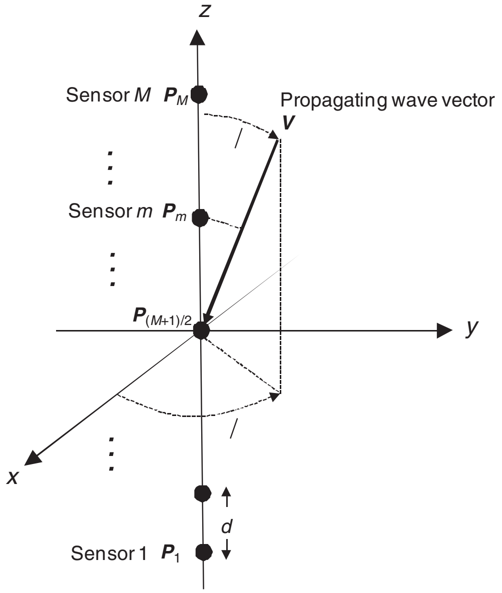


Figure 17.29 A directional plane wave propagating in the direction (θ, ϕ) towards a linear array of M sensors, uniformly spaced with a distance of d between adjacent sensors.

Assume there are an odd number of sensors spaced uniformly along the z-axis (zenith axis) with the middle sensor positioned at the origin of the co-ordinate system. In Figure 17.29, a vector $\boldsymbol{V}$ is shown propagating in the spherical direction with a zenith angle of θ and azimuth of angle ϕ. Assuming that the signal is a monochromatic (in this context this means single frequency) wave, we have

$$V = \frac{2\pi}{\lambda} a(\theta, \phi) = \frac{2\pi f}{c} a(\theta, \phi) \tag{17.70}$$

where $\lambda = \frac{c}{f}$ is the wavelength of the signal of frequency f Hz travelling at speed of c m/s. Note that it is assumed that the source is located in the far-field, so that its signal propagates towards microphones as a plane wave.

17.5.1.2 Directional of Arrival (DoA) and Time of Arrival (ToA)

The difference in time of arrival (ToA) of a plane wave, travelling along the direction (θ, ϕ), at two successive sensors say the m^{th} and the $m+1$ sensors, $\tau_{m,m+1}$, is the ratio of the extra distance between the sensors that the wave has to travel to the speed of the wave c. Note in the case of the arrangement shown in Figure 17.29, where the sensors are placed in the vertical z-direction, the time delay $\tau_{m,m+1}$ is given by

$$\tau_{m,m+1} = \frac{d\cos(\theta)}{c} \tag{17.71}$$

If the m^{th} sensor is at position vector $\boldsymbol{P}_m$ relative to the origin of the coordinate system, see Figure 17.29, then the time difference between the arrival of the waveform at sensor m and arrival at the origin of the coordinate system is given by

$$\tau_m = \frac{P_m a(\theta, \phi)}{c} \tag{17.72}$$

where $P_m a(\theta, \phi)$ is the projection of the position vector of the m^{th} sensor P_m along the vector of the direction of travel of the wave $a(\theta, \phi)$; it represents the distance that the wave has to travel between the m^{th} sensor and the origin of the coordinate system.

17.5.1.3 Steering the Array Direction: Equalisation of the ToAs at the Sensors

The steering of a microphone array is achieved by changing the phase (delay) of the signals received at the sensors. An array of sensors is steered towards a particular direction in space, e.g. the spherical coordinates (θ, ϕ), if the sensor signals from that direction are time-aligned. The time alignment of the sensors' signals may be achieved by adaptive filtering and/or delaying of the sensors output. The summation of the sensors' outputs reinforce the time-aligned waveforms from the steered direction whereas the waveforms from other directions will not be in-phase, will not combine constructively and would be relatively attenuated. The amount of attenuation depends on the number of sensors, their configuration, spacing and the frequency of the incoming waves.

In the frequency domain, the signal received at each sensor can be represented as the product of the discrete Fourier transform (DFT) of the incoming wave $X(f, \theta)$ and a multiplicative term that models the frequency response of the channels between the source and the sensors. Hence, the set of the received signals at the M sensors can be modelled as

$$X(f,\theta)[A_1(f,\theta)e^{-j2\pi f\tau_1(\theta)}, \ldots, A_M(f,\theta)e^{-j2\pi f\tau_M(\theta)}] \tag{17.73}$$

Where $\Lambda_k(f,\theta), \tau_k(\theta)$ and $e^{-j2\pi f\tau_k(\theta)}$ are respectively the amplitude, delay and phase changes that the signal undergoes in its propagation to the k^{th} sensor. The sum of the received signals is given by

$$Y(f,\theta) = X(f,\theta)\sum_{k=1}^{M} A_k(f,\theta)e^{-j2\pi f\tau_k(\theta)} \tag{17.74}$$

The frequency response of the array is given by

$$H(f,\theta) = \sum_{k=1}^{M} A_k(f,\theta)e^{-j2\pi f\tau_k(\theta)} \tag{17.75}$$

Now assume the signal received by the k^{th} sensor is processed by a finite impulse response (FIR) filter with an impulse response of $w_k(m)$ and a frequency response of $W_k(f) = |W_k(f)|e^{j\varphi_k(f)}$, where $|W_k(f)|$ and $e^{j\varphi_k(f)}$ are the magnitude and phase terms of the k^{th} filter response. Note for a delay of $\tau_k(\theta)$, the phase is $\varphi_k(f) = 2\pi f\tau_k(\theta)$.

The set of M processed sensors' signals can be represented as

$$X(f,\theta)[W_1(f)A_1(f,\theta)e^{-j2\pi f\tau_1(\theta)},\ldots,W_M(f)A_M(f,\theta)e^{-j2\pi f\tau_M(\theta)}] \tag{17.76}$$

The sum of the filtered outputs of the sensors can be expressed as

$$Y(f,\theta)=X(f,\theta)\sum_{k=1}^{M}W_k(f)A_\kappa(f,\theta)e^{-j2\pi f\tau_k(\theta)} \tag{17.77}$$

Equation (17.77) can be rewritten in terms of magnitude and phase of the filter as

$$Y(f,\theta)=X(f,\theta)\sum_{k=1}^{M}|W_k(f)|A_k(f,\theta)e^{-j2\pi f\tau_k(\theta)}e^{-j\varphi_k(f)} \tag{17.78}$$

Note that the magnitude of the frequency response of the combined array and filter system is $|W_k(f)|A_k(f,\theta)$ and the combined phase term is $e^{-j2\pi f\tau_k(\theta)-j\varphi_k(f)}$. Note for the sensors' signals to add constructively all the phases $e^{-j2\pi f\tau_k(\theta)-j\varphi_k(f)}$ for different sensors need to be equal.

17.5.1.4 The Frequency Response of a Delay-Sum Beamformer

Sound waves arriving at the microphone array, Figure (17.30), at an incident angle of θ reach successive microphones at different times due to the different distances that the waves have to travel. The additional

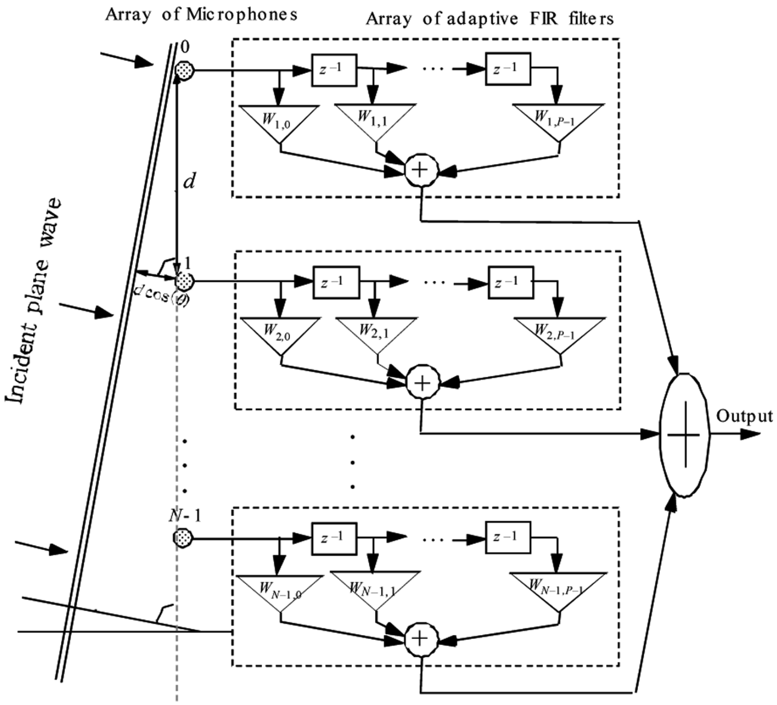


Figure 17.30 Illustration of beam-forming. The array of filters can be adjusted to change the 'look' direction (θ) of the beam.

distance that a wave has travelled to reach an adjacent microphone is $d\ \cos(\theta)$. Note that if $\theta = 90^\circ$ 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 with a speed of $c = 342$ metres per second at a room temperature of 25 °C. The additional time taken for sound to travel a distance of $d\cos(\theta)$ metres to reach an adjacent microphone $\tau_{m,m+1} = \frac{dcos(\theta)}{c}$ as expressed in Equation (17.71).

For a sine wave it is easy to see that if the distance between two adjacent microphones and the angle of incidence of the wave θ is such that it results in a time delay of arrival of sound 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.

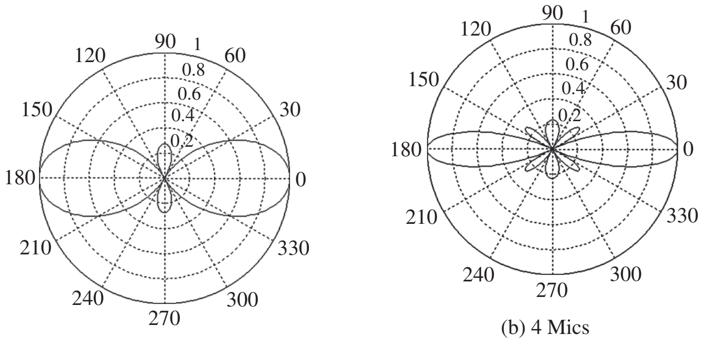
The combined output of N microphones can be expressed as

$$y(m) = \sum_{i=0}^{N-1} x(m - \tau_i) \tag{17.79}$$

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

$$\begin{aligned} H(f,\theta) &= \sum_{m=0}^{N-1} e^{-j2\pi fm\tau} \\ &= \sum_{m=0}^{N-1} e^{-j2\pi fm\left(\frac{d\cos(\theta)}{c}\right)} \end{aligned} \tag{17.80}$$

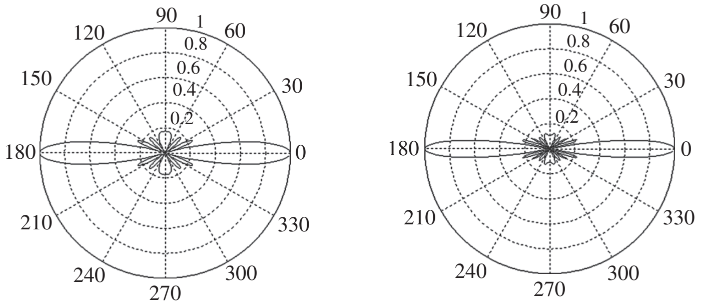
Equation (17.80) is used to plot the response of a microphone array for a selected frequency and varying direction of arrival, Figure 17.31. The angle of arrival can be changed or adapted using the adaptive filters shown in Figure 17.30. Note that the bandwidth of the beam decreases with the increasing number of microphones but it is also a function of the distance between the microphones and the frequency of the sound.



(a) 2 Mics

(b) 4 Mics

Figure 17.31 Illustration of the beam pattern of a microphone array, in response to a frequency of 2 kHz with the microphones spaced at $d = 10$ cm. The number of microphones are varied.



(c) 6 Mics

(d) 8 Mics

Figure 17.31 (*continued*).

17.6 Speech Distortion Measurements

This section describes methods for measuring the quality of speech degraded by noise. Speech quality measures have a number of important uses including:

(1) When evaluated off-line they can be used to compare an algorithm with alternatives or to evaluate the use of different probability models within an algorithm.
(2) When evaluated in real time they can be used to select the best parameter configuration, codebooks or posterior probability from a number of choices.
(3) They can provide the user/operator with an online report of the communication system performance.

There are two types of distortion and quality measures: objective measures such as signal-to-noise ratio and subjective measures such as mean opinion score. The most commonly used distortion measures for speech signals are the following.

17.6.1 Signal-to-Noise Ratio – SNR

The most commonly used measure for quality of speech is signal-to-noise ratio (SNR). An average SNR measure is defined as

$$SNR = 10\log_{10}\left(\frac{P_{Signal}}{P_{Noise}}\right)\text{dB} \tag{17.81}$$

where P_{Signal} and P_{Noise} are the power of signal and noise respectively.

17.6.2 Segmental Signal to Noise Ratio – SNR$_{\text{seg}}$

The segmental SNR is defined as

$$SNR_{seg} = \frac{1}{K}\sum_{k=0}^{K-1} 10\ \log_{10}\left(\sum_{m=0}^{N-1} x_k^2(m) \Bigg/ \sum_{m=0}^{N-1}\left(x_k(m) - \hat{x}_k(m)\right)^2\right)\text{dB} \tag{17.82}$$

where $x_k(m)$ and $\hat{x}_m(m)$ 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 about the average SNR.

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.

17.6.3 Itakura–Saito Distance – ISD

The Itakura–Saito Distance (ISD) measure is defined as

$$ISD_{12} = \frac{1}{N}\sum_{j=1}^{N}\frac{(\boldsymbol{a}_1(j)-\boldsymbol{a}_2(j))^T\boldsymbol{R}_1(j)(\boldsymbol{a}_1(j)-\boldsymbol{a}_2(j))}{\boldsymbol{a}_1(j)\boldsymbol{R}_1(j)\boldsymbol{a}_1(j)^T} \tag{17.83}$$

where $\boldsymbol{a}_1(j)$ and $\boldsymbol{a}_2(j)$ are the linear predication model coefficient vectors calculated from clean and transformed speech at frame j and $\boldsymbol{R}_1(j)$ is an autocorrelation matrix derived from the clean speech. Due to the asymmetry of ISD measure (i.e. $\mathrm{ISD}_{21} \neq \mathrm{ISD}_{12}$) the following segmental ISD measure is used

$$ISD_{sym} = (ISD_{12} + ISD_{21})/2 \tag{17.84}$$

The ISD criterion is a more balanced measure of the distance between an original clean speech signal and a distorted speech signal compared with the SNR measures of Equations (17.81)–(17.82).

17.6.4 Harmonicity Distance – HD

To measure the distortions of the harmonic structure of speech, a harmonic contrast function is defined as

$$HD = \frac{1}{NH \times N_{frames}}\sum_{N_{frames}}\sum_{k=1}^{NH} 10\log\frac{P_k + P_{k+1}}{2P_{k,k+1}} \tag{17.85}$$

where P_k is the power at harmonic k, $P_{k,k+1}$ is the power at the trough between harmonics k and $k+1$, NH is the number of harmonics and N_{frames} is the number of speech frames.

17.6.5 Diagnostic Rhyme Test – DRT

The diagnostic rhyme test consists of 96 word pairs which differ by a single acoustic feature in the initial consonant. Word pairs are chosen to evaluate the six phonetic characteristics listed in Table 17.5. The listener hears one word at a time and marks the answering sheet to indicate which of the two words she or he thinks is the correct one. The results are summarised by averaging the error rates from all the answers. Usually, only the total error rate percentage is given, but also single consonants and how they are confused with each other can be investigated with confusion matrices.

Table 17.5 DRT characteristics.

Characteristics	Description	Examples
Voicing	voiced – unvoiced	veal – feel, dense – tense
Nasality	nasal – oral	reed – deed
Sustention	sustained – interrupted	vee – bee, sheat – cheat
Sibilation	sibilated – unsibilated	sing – thing
Graveness	grave – acute	weed – reed
Compactness	compact – diffuse	key – tea, show – sow

DRT provides useful diagnostic information on recognition of the initial consonant. However, it does not test vowels or prosodic features and hence it is not suitable for evaluation of the overall quality of speech. A further deficiency is that the test material is rather limited and the test items do not occur with equal probability, hence the method does not test all possible confusions between various consonants.

17.6.6 Mean Opinion Score – MOS

The mean opinion score provides a numerical evaluation of the perceived quality of speech. The MOS, as shown in Table 17.6, is expressed as a number in the range 1 to 5, where 1 is the lowest perceived quality, and 5 is the highest perceived quality. MOS can be used to assess the perceived qualities of the inputs and outputs of a communication channel or a speech enhancement method.

Table 17.6 Mean Opinion Score (MOS).

MOS	Quality	Distortion/Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

MOS tests for voice are specified by ITU-T recommendation P.800. The MOS is obtained by averaging the results of a set of standard, subjective tests where a number of listeners rate the quality of test speech sentences read by male and female speakers. Each listener gives each sentence a rating using the rating scheme in Table 17.6.

The MOS is the mean of all the individual test scores. Note that one could ask the listening subjects to rate the quality or impairment (degradation) of speech.

17.6.7 Perceptual Evaluation of Speech Quality – PESQ

PESQ stands for ‘perceptual evaluation of speech quality’. PESQ is an enhanced perceptual quality measurement for voice quality in telecommunications. PESQ was specifically developed to be applicable to end-to-end voice quality testing under real network conditions, such as VoIP, POTS, ISDN, GSM etc. The PESQ algorithm is designed to predict subjective opinion scores of a degraded speech sample. PESQ returns a score from 4.5 to −0.5, with higher scores indicating better quality.

PESQ is officially approved as ITU-T recommendation P.862. PESQ was developed by KPN Research in the Netherlands and British Telecommunications (BT). PESQ analyses specific parameters of the speech signal, including time warping, variable delays, transcoding, and noise. It is primarily intended for applications in codec evaluation and network testing but is now increasingly used in speech enhancement. It is best to use PESQ as an addition to other methods such as mean opinion scores (MOS), where an average is taken of the subjective opinion of human listeners.

Bibliography

Appendix I. (1999) A High Quality Low-Complexity Algorithm for Packet Loss Concealment with G.711, ITU-T Recommend. G.711, Sept.

Bakamidis S., Dendrinos M. and Carayannis G. (1991) Speech Enhancement From Noise: A Regenerative Approach. *Speech Communication*, **10**: 44–57.

Benesty J., Chen J., and Huang Y., (2008) *Microphone Array Signal Processing*. Springer-Verlag, Germany.

Berouti M., Schwartz R. and Makhoul J. (1979) Enhancement of Speech Corrupted by Acoustic Noise, *Proc. IEEE* Int. *Conf. Acoust., Speech, Signal Process*: 208–211, Apr.

Boll S.F. (1979) Suppression of Acoustic Noise in Speech Using Spectral Subtraction. *IEEE Trans. Acoust., Speech and Signal Proc*., **ASSP-27**: 113–120.

Cohen I. (2004) Speech Enhancement Using a Noncausal A Priori SNR Estimator, *IEEE Signal Processing Letters*, **11**(9): 725–728.

Deller J.R., Proakis J.G. and Hansen J.H. (1993) *Discrete-Time Processing of Speech Signals*. New York: Macmillan Publishing Company.

Elsabrouty M., Bouchard M., Aboulnasr T. (2003), A new hybrid longterm and short-term prediction algorithm for packet loss erasure over IP-networks, in *Proc. 7th Int. Symp. Signal Processing and Its Applications*, **1**: 361–364.

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 Processing*, **ASSP-33**: 443–445.

Ephraim Y. and Van Trees H.L. (1995) A Signal Subspace Approach for Speech Enhancement, *IEEE Trans. Speech and Audio Processing* **3**(4): 251–266.

Esquef P.A. and Biscainho L.W.P. (2006) An Efficient Model-Based Multirate Method for Reconstruction of Audio Signals Across Long Gaps, *IEEE Trans. Audio, Speech and Language Processing*, **14**(4): 1391–1400.

Gannot S., Burshtein D. and Weinstein E. (1998) Iterative and Sequential Kalman Filter-based Speech Enhancement, *IEEE Trans. Speech and Audio Processing*, **6**(4): 373–385.

Goodman D.J., Lockhart G.B., O.J. Wasem, and W.C. Wang (1986), ‘Waveform substitution techniques for recovering missing speech segments in packet voice communications,’ *IEEE Trans. Acoustics, Speech, Signal Processing*, **ASSP-34**(5): 1440–1448, Oct..

Griffin D.W. and Lim J.S. (1988) Multiband-excitation vocoder, *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-36**(2): 236–243.

Gündüzhan E., Momtahan K. (2001), A linear prediction based packet loss concealment algorithm for PCM coded speech, *IEEE Trans. Speech Audio Process*., **9**(6): 778–785, Nov.

Hansen J.H.L. and Clements M.A. (1987) Iterative Speech Enhancement with Spectral Constraints, *Proc. of ICASSP*: 189–192.

Janssen A.J.E.M., Veldhuis R., Vries L.B. (1986), ‘Adaptive interpolation of discrete-time signals that can be modelled as ar processes’. *IEEE Trans. Acoustics, Speech and Signal Processing*, **ASSP-34**(2): 317–330, April.

Kalman R. (1960) A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME, Journal of Basing Engineering*, **82**: 34–35.

Kauppinen I., Roth K. (2002), Audio signal restoration–theory and applications’, in *Proc. 5th Int. Conf. on Digital Audio Effects*, (Ham-burg, Germany): 105–110, Sept..

Kondoz A.M. (1999), *Digital Speech: coding for low bit rate communication systems*. John Wiley & Sons, Ltd.

Lim J.S. and Oppenheim A.V. (1978) All-pole Modelling of Degrade Speech, *IEEE Trans. Acoust. Speech, Signal Processing* **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, Dec.

Lindblom J., Hedelin P., Packet loss concealment based on sinusoidal extrapolation, in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing 2002*, Orlando, FL, USA, May 2002, **1**: 173–176.

Lindblom J. and Hedelin P. (2002) Packet Loss Concealment Based on Sinusoidal Modelling, *Proc. IEEE Workshop on Speech Coding*, Ibaraki, Japan, October: 65–67.

Mack G.A. and Jain V.K. (1985) A Compensated-Kalman Speech Parameter Estimator, **ICASSP**: 1129–1132.

Martin R. (2002) Speech Enhancement Using MMSE Short Time Spectral Estimation with Gamma Distributed Speech Priors, *IEEE* **ICASSP’02**, Orlando, FL, May.

Martin R. and Breithaupt C. (2003) Speech Enhancement in the DFT Domain Using Laplacian Speech Priors, *Proc. Int. Workshop Acoustic Echo and Noise Control* (**IWAENC**): 87–90.

McAulay R.J. and Malpass M.L. (1980) Speech Enhancement Using a Soft-Decision Noise Suppression Filter, *IEEE Trans. Acoust., Speech, Signal Processing*, **ASSP-28**: 137–145.

Mccandles S.S. (1974) An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra, *IEEE Trans. Acoustics, Speech and Signal Processing* **22**: 135–141.

Milner B.P. and James A.B. (2004) An Analysis of Packet Loss Models for Distributed Speech Recognition, *Proc. ICSLP 2004*: 1549–1552.

De Moor B. (1993) The Singular Value Decomposition and Long and Short Spaces of Noisy Matrices, *IEEE Trans. Signal Processing*, **41**(9): 2826–2838.

Murthi M.N., Rødbro C.A., Andersen S.V. and Jensen S.H. (2006) Packet Loss Concealment with Natural Variations Using HMM, *ICASSP 2006*, **1**: I-21–24.

Paliwal K.K. and Alsteris L.D. (2003), Usefulness of phase spectrum in human speech perception, Proc. European Conf. Speech Communication and Technology, *EUROSPEECH-03*, Geneva, Switzerland: 2117–2120, Sept. 2003.

Paliwal K.K. and Basu A. (1987) A Speech Enhancement Method Based on Kalman Filtering, *Proc. of ICASSP*: 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 Schafer R.W. (1978) *Digital Processing of Speech Signals*. Prentice Hall, Englewood Cliffs, NJ.

Rabiner L.R. and Juang B.H. (1993) *Fundamentals of Speech Recognition*. 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, *Proc. of Eurospeech*: 2405–2408.

Rigoll G. (1986) A New Algorithm for Estimation of Formant Trajectories Directly from the Speech Signal Based on an Extended Kalman-filter, *Proc. of ICASSP*: 1229–1232.

Rødbro C.A., Christensen M.G., Andersen S.V. and Jensen S.H. (2003) Compressed Domain Packet Loss Concealment of Sinusoidally Coded Speech, *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc.*, **1**: 104–107.

Rødbro C.A., Murthi M.N., Andersen S.V. and Jensen S.H. (2005) Hidden Markov Model-based Packet Loss Concealment for Voice Over IP, *IEEE Trans. Audio, Speech, and Language Processing*, **PP (99)**: 1–17.

Sameti H., Sheikhzadeh H., Deng L. and Brennan R.L. (1998) HMM-Based Strategies for Enhancement of Speech Signals Embedded in Non-Stationary Noise. *IEEE Trans. Speech and Audio Processing*, **6**(5): 445–455.

Schroeder M.R. (1999) *Computer Speech: Recognition, Compression*. Synthesis, Springer.

Shafer R.W. and Rabiner L.R. (1970) System for Automatic Formant Analysis of Voiced Speech, *J. Acoust. Soc. Am.* **47**(2): 634–650.

Stylianou Y. (2001), 'Applying the harmonic plus noise model in concatenative speech synthesis', *IEEE Trans. Speech and Audio Processing*, **9**(1): 21–29, Jan.

Valenzuela R.A. and. Animalu C. N (1989), 'A new voice-packet reconstruction technique,' in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*: 1334–1336.

Vaseghi S., Zavarehei E. and Yan Q. (2006) Speech Bandwidth Extension: Extrapolations of Spectral Envelope and Harmonicity Quality of Excitation, *ICASSP 2006*, **3**: III-844–847.

Vaseghi S. and Rayner P.J.W. (1990), 'Detection and suppression of impulsive noise in speech communication systems', *IEE Proc.*, Part 1, **137**(1): 38–46, February.

Wang J., Gibson J.D. (2001), 'Parameter interpolation to enhance the frame erasure robustness of CELP coders in packet networks,' in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, **2**: 745–748.

Wang, D.L. and Lim, J. *S. (1982) The Unimportance of Phase in Speech Enhancements, IEEE Trans. Acoust., Speech and Signal Processing*, **30**: 679–681.

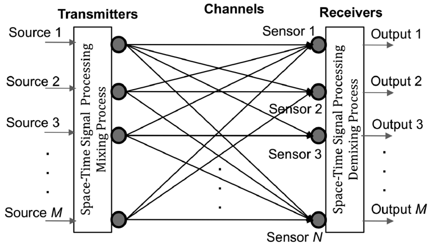
Wang J., Gibson J.D. (2000), 'Performance comparison of intraframe and interframe LSF quantization in packet networks', *IEEE Workshop on Speech Coding*: 126–128.

Weber K., Bengio S. and Bourlard H. (2001) HMM2-Extraction of Formant Structures and Their use for Robust ASR, *Proc. of Eurospeech*: 607–610.

Yan Q. and Vaseghi S. (2003) Analysis, Modelling and Synthesis of Formants of British, American and Australian Accents, *IEEE International Conference on ICASSP Acoustics, Speech, and Signal Processing* **1**(6–10): I-712–I-715, April.

Zavarehei E., Vaseghi S. and Yan Q., (2007) Noisy Speech Enhancement Using Harmonic-Noise Model and Codebook-Based Post-Processing', *IEEE Trans. Audio, Speech, and Language Processing*, **15**(4): 1194–1203.

Zavarehei E., Vaseghi S. (2008) Interpolation of Lost Speech Segments Using LP-HNM Model With Codebook Post-Processing, Multimedia, *IEEE Trans.*, **10**(3): 493 –502.



18

Multiple-Input Multiple-Output Systems, Independent Component Analysis

Multiple-input multiple-output (MIMO) signal processing systems are employed in a wide range of applications including multi-sensors biological signal processing systems, phased-array radars, steerable directional antenna arrays for mobile phone systems, microphone arrays for speech enhancement and multichannel audio entertainment systems.

The aim of MIMO signal processing is to attain one of the following two objectives: (1) signal separation; i.e. to separate a number of independent signals and noise given an observation of a mix of the signals and noise and (2) signal combination; i.e. to mix a number of signals to achieve some desired property such as forming a narrow transmission or reception beam.

In this chapter we introduce MIMO systems and consider independent component analysis (ICA) for separation of signals in MIMO systems. Although ICA has some common features with beam-forming, it is also fundamentally different, in that ICA can separate independent signals propagating together in the same direction and space whereas beam-forming separates signals in terms of their direction of propagation in the space.

18.1 Introduction

MIMO signal processing refers to a wide range space-time signal processing systems. In a typical multiple input multiple output (MIMO) transmitter/receiver system, shown Figure 18.1, there are M independent sources of signals and noise dispersed or configured in some fashion in space and there is an array of N sensors usually positioned in some optimally designed configuration. Each sensor receives (or transmitter transmits) a different mixture or combination of the source signals.

The overall aim of MIMO signal processing is to attain one or more of the following objectives:

(1) Noise reduction – For example, as in antenna arrays for radio waves, microphone arrays for audio waves or arrays of ECG (electrocardiography) sensors for heartbeats or EEG (Electroencephalography) signals for brain waves. MIMO signal processing can be used to enhance the signal to noise ratio of the received signals.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

(2) Diversity gain – This is the use of MIMO systems, or multipath effect, to utilise the arrivals of multiple versions of a signal at the reievers in order to enhance the signal to noise ratio.
(3) Multiple signal recording and separation – For example in EEG the aim is to separate the signal activities of different parts of the brain recorded by an array of sensors.
(4) Adaptive steerable beam-forming communication – This allows an array of transmitters/receivers to form a narrow beam to target and adaptively track a receiver or transmitter.
(5) Capacity increase in mobile communication – The use of space-time MIMO signal processing methods allows improvement in the capacity and the quality of communication.

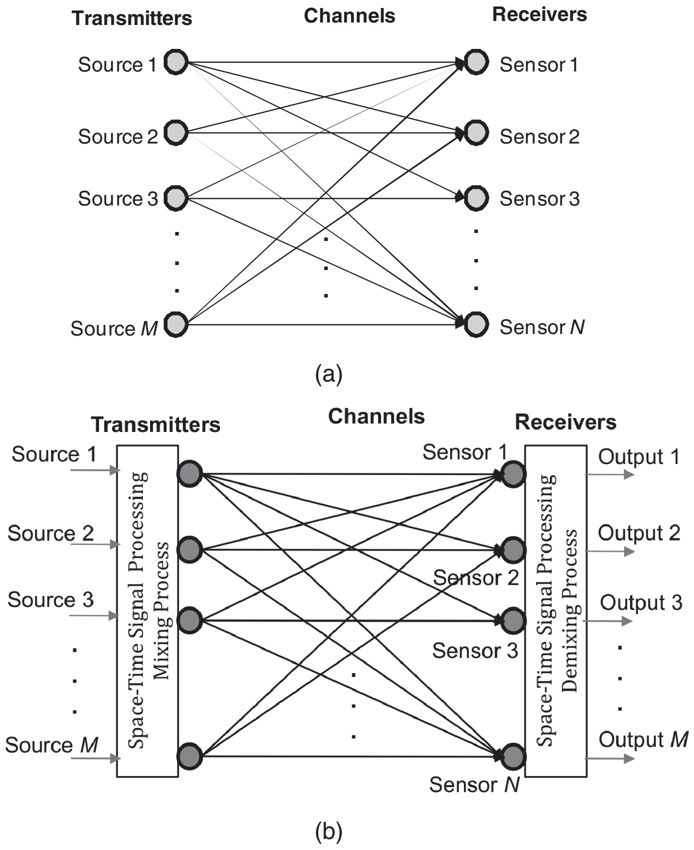


Figure 18.1 (a) Illustration of a MIMO system with M sources and N sensors, (b) a MIMO system with space-time signal processing for mixing the signals at the transmitter and and demixing the signals at the receiver.

The functions of MIMO signal processing systems can be broadly categorised into two classes:

(1) MIMO transmission – These usually involve a method of mixing of the source signals prior to transmission. Examples include the various methods of multiplexing of signals in telecommunications and beam-forming; the purpose of beam-forming is to direct and confine the space of propagation of the signals into a narrow beam.

(2) MIMO reception – These usually involve a method of separation of received signals; i.e. the de-mixing and extraction of the individual source signals from the mixed observation. In beam-forming, the signal de-mixing process consists of directing a narrow-beam sensors array at each source with the effect of screening-out other sources of signals and noise. In independent component analysis the objective is to separate the mixed signals.

In the signal de-mixing problem the parameters of the equations that describe the relationships between the source signals and the observations are usually unknown and need to be estimated using a process referred to as blind source separation (BSS). The most popular method for blind source separation is the independent component analysis (ICA). ICA involves the determination of a de-mixing matrix that transform the observation signal into a set of independent source signal components. This is often achieved by diagonalising the second order statistic (i.e. covariance) and the higher order statistics usually the fourth order statistic (i.e. kurtosis) of the signal. ICA is often used as an add-on to principal component analysis (PCA) for the processing of non-Gaussian signals such as speech, image and biomedical signals.

18.2 A note on comparison of beam-forming arrays and ICA

Beam-forming and ICA are often used in MIMO systems either individually or combined. A beam-forming array separates signals spatially; however, independent signals propagating along the same direction would not be separated by a beam former. In contrast, ICA separates independent signals arriving at a sensor array irrespective of their directions of arrivals.

18.3 MIMO Signal Propagation and Mixing Models

This section describes three linear signal propagation and mixing models, describing the relationship between an M-dimensional input signal vector $\boldsymbol{s}$ and N-dimensional output signal vector $\boldsymbol{x}$ in a MIMO communication system. These commonly used propagation models are:

(1) The *instantaneous* propagation and mixing model, where it is assumed that the differences in propagation delays are negligible and the signal propagation from a source i to a sensor j is modelled by a single attenuation factor a_{ij}.
(2) The *anechoic* (direct path) propagation and mixing model, where the propagation from a source i to a sensor j is modelled by a direct path with two parameters; an attenuation factor a_{ij} and a time delay τ_{ij}.
(3) The *echoic* (convolutional) propagation and mixing model where the direct and non-direct propagation paths from a source i to a sensor j is modelled by the impulse response coefficients of a linear filter $\boldsymbol{h}_{ij}$

These signal mixing models are described next.

18.3.1 Instantaneous Mixing Models

The instantaneous signal mixing model assumes that all signals arrive at the sensors at the same time, i.e. without relative delays. The instantaneous mixing model is particularly useful in applications where the differences in propagation delays between the different signal sources and sensors are negligible in comparison to the sampling period, this may be due to a combination of low signal frequencies and relatively short and similar distances between the sources and sensors and/or relatively fast propagation speed. For example, in biomedical applications when multiple sensors are used for the recording of the electrical activities of heart or brain, i.e. Electrocardiograph or Electroencephalography signals, the sensors are relatively close to each other and the propagation speeds are relatively high compared to the signal frequency and sampling frequency.

The instantaneous mixing model describing the relationship between the source signal vector $\boldsymbol{s} = [s_1(m), s_2(m), \ldots, s_M(m)]$ and the observation signal vector $\boldsymbol{x}(m) = [x_1(m), x_2(m), \ldots, x_N(m)]$, can be described as

$$x_i(m) = a_{i1}s_1(m) + a_{i2}s_2(m) + \cdots + a_{iN}s_N(m) \qquad i = 1, \ldots, M \tag{18.1}$$

In compact notation Equation (18.1) can be written as

$$\boldsymbol{x}(m) = \boldsymbol{A}\boldsymbol{s}(m) \tag{18.2}$$

Where $\boldsymbol{A}$ is an $N \times M$ dimensional mixing matrix that transforms the M input sources to the N output observations. The output vector $\boldsymbol{x}(m)$ can be expressed as a weighted combination of the M column vectors of $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_M]$ as

$$\boldsymbol{x}(m) = \boldsymbol{a}_1 s_1(m) + \boldsymbol{a}_2 s_2(m) + \cdots + \boldsymbol{a}_M s_M(m) \tag{18.3}$$

where $\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iN}]^{\mathrm{T}}$ is the N-dimensional i^{th} column of $\boldsymbol{A}$.

Note that all linear filtering and signal transformation methods can be cast in the form of Equations (18.1)–(18.3). The matrix $\boldsymbol{A}$ is the system matrix that transforms the input signal vector $\boldsymbol{x}$ to the output signal vector $\boldsymbol{y}$, hence the analysis of the structure of $\boldsymbol{A}$ and development of methods for efficient representation and application of $\boldsymbol{A}$ are of particular interest in signal processing and system analysis.

The ideal demixing matrix is the inverse of the mixing matrix $\boldsymbol{A}$, from which the source signal can be obtained as

$$\boldsymbol{s}(m) = \boldsymbol{A}^{-1}\boldsymbol{x}(m) \tag{18.4}$$

However, in the blind source separation problem, where all that is available is the observation signal $\boldsymbol{x}$, it is not normally possible to exactly estimate the mixing matrix $\boldsymbol{A}$ or it's inverse (i.e. demixing matrix) $\boldsymbol{A}^{-1}$. This is because, as it will be shown later in this chapter, a scaling of the source vector **s** or a change of the order of the elements of the source vector **s** will have no effect on the observation vector $\boldsymbol{x}$ provided that the estimate of the matrix $\boldsymbol{A}$ is also modified appropriately to cancel out the effects of these changes. Hence, we cannot identify the scale of the sources or their order in the vector. Therefore in practice the demixing operation will be expressed as

$$\boldsymbol{y}(m) = \boldsymbol{B}\boldsymbol{x}(m) \tag{18.5}$$

where the matrix $\boldsymbol{B}$ is ideally the inverse of the matrix $\boldsymbol{A}$, but more realistically the relationship between the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ may be expressed as

$$\boldsymbol{B} = \alpha\boldsymbol{P}\boldsymbol{A} \tag{18.6}$$

where the scale factor α and the permutation matrix $\boldsymbol{P}$ are unknown.

18.3.2 Anechoic, Delay and Attenuation, Mixing Models

An anechoic environment is an echo-free and reflection-free propagation environment. Hence, in an anechoic environment the signal from a source reaches each sensor via the direct propagation path only which may be modelled by two parameters: an attenuation factor a and a propagation time delay τ.

Whereas in the instantaneous model the relationship, between the signal emitted from the source i and the same signal received at the sensor j, is modelled by a single gain/attenuation factor a_{ij}, the anechoic model additionally includes a propagation delay $\tau_{ij} = d/v$ for the time taken for the signal to propagate a distance of d meters at speed of v m/s from source i to reach sensor j. Hence, the observation signal $x_j(m)$ is modelled as

$$x_j(m) = \sum_{i=1}^{M} a_{ij} s_i(m - \tau_{ij}), \qquad j = 1, \ldots, N \tag{18.7}$$

Note that in the discrete-frequency (DFT) domain, where a time delay of τ_{ij} is represented by a multiplicative phase term $e^{-j2\pi k\tau_{ij}}$, the k^{th} discrete frequency component of the j^{th} observation signal may be expressed as

$$X_j(k) = \sum_{i=1}^{M} a_{ij} e^{-j2\pi k\tau_{ij}} S_i(k) \qquad j = 1, \ldots, N \tag{18.8}$$

Equation (18.8) can be written in a matrix form as

$$\boldsymbol{X}(k) = [\boldsymbol{A}(k) \bullet \boldsymbol{\Phi}(k)]\boldsymbol{S}(k) \tag{18.9}$$

Where $\boldsymbol{A}(k)$ is an $N \times M$ dimensional coefficient matrix, $(\boldsymbol{A}(k))_{ij} = a_{ij}, \boldsymbol{\Phi}$ is an $N \times M$ phase matrix, $(\boldsymbol{\Phi}(k))_{ij} = e^{-j2\pi k\tau_{ij}}$ and $\boldsymbol{A}(k) \bullet \boldsymbol{\Phi}(k)$ is the Hadamard matrix product, also known as the entry-wise matrix product, $(\boldsymbol{A}(k) \bullet \boldsymbol{\Phi}(k))_{ij} = a_{ij} e^{-j2\pi k\tau_{ij}}$.

18.3.3 Convolutional Mixing Models

The convolutional propagation model is useful in practical applications where the propagation environment is modelled as echoic, i.e. the model allows for the propagation of the signal via a number of reflection paths in addition to the direct propagation path.

In the convolutive model the propagation path from source i to sensor j is modelled by a filter $a_{ij}(l)$ with an impulse response of length say $L + 1$ taps, hence, the observation signal $x_j(t)$ is modelled as

$$x_j(m) = \sum_{i=1}^{M} \sum_{l=1}^{L} a_{ij}(l) s_i(m - l), \qquad i = 1, \ldots, N \tag{18.10}$$

Note that in the discrete-frequency frequency domain, the k^{th} frequency component of the j^{th} observation signal may be expressed as

$$X_j(k) = \sum_{i=1}^{M} A_{ij}(k) S_i(k), \qquad j = 1, \ldots, N \tag{18.11}$$

Hence in the frequency domain, for each frequency channel k, the convolutional channel model can be expressed as a multiplicative matrix transformation as

$$\boldsymbol{X}(k) = \boldsymbol{A}(k)\boldsymbol{S}(k) \tag{18.12}$$

where the complex frequency vector $\boldsymbol{S}(k) = [S_1(k), \ldots, S_M(k)]^{\text{T}}$ is the M-dimensional input signal vector, $\boldsymbol{X}(k) = [X_1(k), \ldots, X_N(k)]^{\text{T}}$ is the N-dimensional complex frequency output signal vector and $\boldsymbol{A}(k)$, the $N \times M$ dimensional complex matrix, is the channel model. Note that in Equation (18.19) all variables are frequency dependent and hence the channel matrix $\boldsymbol{A}(k)$ needs to be estimated for the set of discrete frequencies spanning the signal bandwidth.

18.4 Independent Component Analysis

Independent component analysis (ICA) is a signal processing method for blind identification of an unknown linear mixing model and its inverse or demixing model. As explained in this chapter, ICA employs non-linear objective criterions in order to identify the unknown linear mixing and demixing models.

ICA is a powerful method for analysis and extraction of non-Gaussian features and for demixing of mixed signals. ICA builds on principal component analysis (PCA). As shown in Figure 18.2 usually in preparation for ICA the signal is sphered (i.e. whitened and its eigenvalues equalised). The signal sphereing is done by multiplying the signal vectors by the matrix $\boldsymbol{\Lambda}^{-0.5}\boldsymbol{U}^T$ where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues and $\boldsymbol{U}$ is the matrix of eigen vectors.

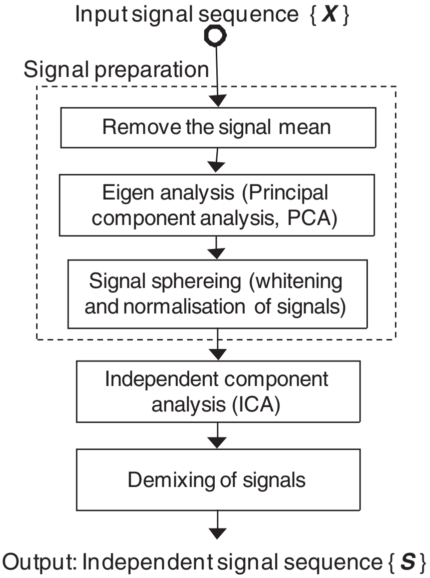

Input signal sequence {X}
Signal preparation
Remove the signal mean
Eigen analysis (Principal component analysis, PCA)
Signal sphereing (whitening and normalisation of signals)
Independent component analysis (ICA)
Demixing of signals
Output: Independent signal sequence {S}

Figure 18.2 Illustration of the main stages (including the preparatory PCA) in ICA.

However, whereas PCA works on the assumed Gaussian structure of the signal and decorrelates the second order statistics (i.e. covariance matrix) of the signal, ICA works on the non-Gaussian structures of the signal and decorrelates the second order, and most importantly, the higher order statistics (usually the fourth order i.e. kurtosis) of the signal process. Since ICA assumes the signals are non-Gaussian the solutions involve the optimisation of nonlinear contrast (objective) functions.

ICA is also known as 'blind' signal separation (BSS) since it requires no knowledge of the signal mixing process other than the assumption that the source signals are statistically independent and that at most no more than one source signal is Gaussian. Some argue that it's more accurate to describe ICA as a tool or a method for solving the general problem of BSS. The ICA works on the principal of calculating a demixing matrix whose output vector would have statistically independent components.

Whereas the classical linear signal processing theory often assumes that the signals are Gaussian distributed processes and hence do not possess any higher order statistics above covariance, ICA works on the assumption that the signals are non-Gaussian and possess higher-order statistics.

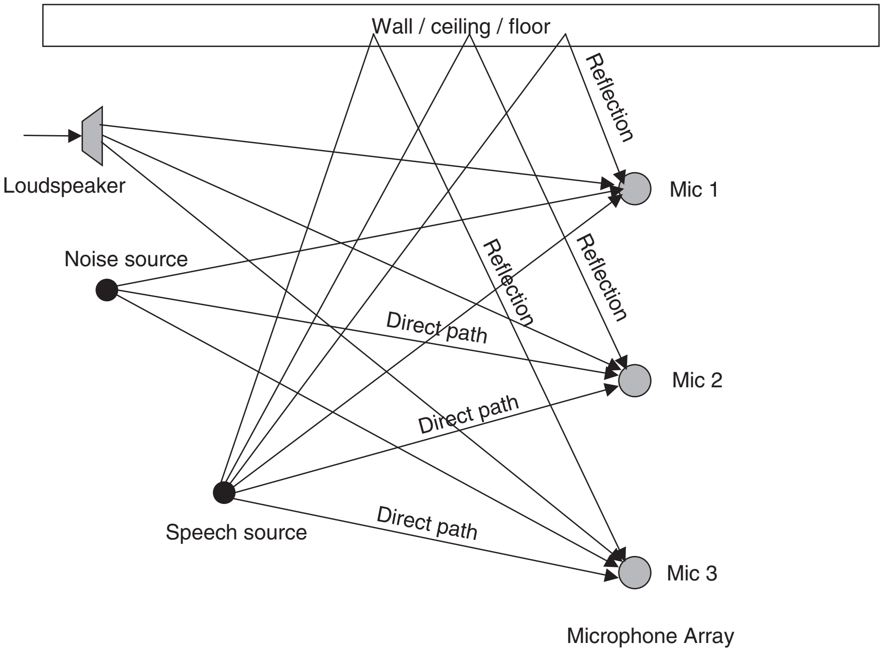


Figure 18.3 Illustration of different sounds and noise arriving at microphones, via direct line of sight paths and via reflections from walls/ceiling/floor.

ICA is primarily used for separation of mixed signals in multi-source multi-sensor applications such as in electrocardiography (EEC) and electroencephalogram (EEG). ICA is also applied in multiple-input multiple-output (MIMO) mobile communication and multiple sensor signal measurement systems where there are N signal sources or transmitters and M sensors or receivers. For example ICA is used in beam-forming in array signal processing applications such as microphone array for directional reception of audio and speech signals (Figure 18.3) and antenna array for directional reception or demixing of electromagnetic radio waves.

The most common examples of applications of ICA include:

(1) Extraction of M sources from an array of N sensors ($N \geq M$) in biomedical signal processing such as electrocardiograph or electroencephalogram signal measurements.
(2) An array of N microphones in a room or car cabin with M sources of sounds and noise.
(3) An array of M radio transmitter antennas and N receiver antennas in beam-forming smart antennas for mobile phones.
(4) Beamforming and array signal processing in radar and sonar.
(5) Image/speech feature extraction and image/speech denoising.

18.4.1 A Note on Orthogonal, Orthonormal and Independent

It’s useful here to define the terms orthogonal, orthonormal and independent. Orthogonal implies that two vectors are geometrically perpendicular or statistically uncorrelated in terms of the second order statistics. Orthonormal vectors are orthogonal and have magnitudes of one. It should be noted that there is a difference between the terms uncorrelated and independent. Independence is a more strict concept and requires the signals to be probabilistically independent and hence uncorrelated not only in the second order statistics but also in all the higher order statistics, whereas in contrast the terms uncorrelated and orthogonal commonly refer to second order statistics and essentially imply that two vectors have zero correlation or that a vector process has diagonal covariance matrix. This is elaborated later in this chapter in the study of independent component analysis.

18.4.2 Statement of ICA Problem

Let the signal samples from N sources form an N-dimensional source signal vector $\boldsymbol{s}(m) = [s_1(m), s_2(m), \ldots, s_M(m)]$ and assume that the output of the array of N sensors is represented by the vector $\boldsymbol{x}(m) = [x_1(m), x_2(m), \ldots, x_N(m)]$. Assume that each sensor output $x_i(m)$ is a linear combination of the M source signals; that is

$$x_i(m) = a_{i1}s_1(m) + a_{i2}s_2(m) + \cdots + a_{iM}s_M(m) \qquad i = 1, \ldots, N \tag{18.13}$$

In a matrix form Equation (18.13), for each discrete time instance m, Equation (18.42) can be written as

$$\begin{bmatrix} x_1(m) \\ x_2(m) \\ \vdots \\ x_N(m) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NM} \end{bmatrix} \begin{bmatrix} s_1(m) \\ s_2(m) \\ \vdots \\ s_M(m) \end{bmatrix} \tag{18.14}$$

Note that when we have a sequence of K N-dimensional observation vectors, with each observation vector composed of the mixings of M samples from the sensors, equation (18.14) becomes

$$\underbrace{\begin{bmatrix} x_1(0) & x_1(1) & \cdots & x_1(K-1) \\ x_2(0) & x_2(1) & \cdots & x_2(K-1) \\ \vdots & \vdots & \ddots & \vdots \\ x_N(0) & x_N(1) & \ldots & x_N(K-1) \end{bmatrix}}_{\substack{\text{Sequence of } K \text{ observation vectors} \\ \text{dimensions: } N \times K}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NM} \end{bmatrix}}_{\substack{\text{Mixing matrix} \\ \text{dimensions: } N \times M}} \underbrace{\begin{bmatrix} s_1(0) & s_1(1) & \cdots & s_1(K-1) \\ s_2(0) & s_2(1) & \cdots & s_2(K-1) \\ \vdots & \vdots & \ddots & \vdots \\ s_M(0) & s_M(1) & \ldots & s_M(K-1) \end{bmatrix}}_{\substack{\text{Sequence of } K \text{ source vectors} \\ \text{dimensions: } M \times K}} \tag{18.15}$$

In a compact notation Equation (18.14) can be rewritten as

$$\boldsymbol{x}(m) = \boldsymbol{A}\boldsymbol{s}(m) \tag{18.16}$$

The matrix $\boldsymbol{A}$ is known as the mixing matrix or the observation matrix. In general the N sources of signals are independent and this implies that the signal vector **s** has a diagonal covariance matrix and diagonal cumulant matrices and that the elements of the vector **s** have zero mutual information. However, the observation signal vector, $\boldsymbol{x}$, are correlated with each other as in general each observation signal contains a mixture of all the source signals as shown by Equation (18.14). Hence the covariance matrix of the observation signal vector $\boldsymbol{x}$ is a non-diagonal matrix and the elements of $\boldsymbol{x}$ may have non-zero mutual information.

In many practical cases of interest all that we have is the sequence of observation vectors $[x(0), x(1), \ldots, x(K-1)]$, the mixing matrix $\boldsymbol{A}$ is unknown and we wish to estimate a demixing matrix $\boldsymbol{W}$ to obtain an estimate of the original signal sequence $[s(0), s(1), \ldots, s(K-1)]$. The demixing problem is the estimation of a matrix $\boldsymbol{W}$ such that

$$\hat{\boldsymbol{s}}(m) = \boldsymbol{W}\boldsymbol{x}(m) \tag{18.17}$$

where ideally the demixing matrix $\boldsymbol{W} = \boldsymbol{A}^{-1}$. Note that each element of the source signal $s_i(m)$ is estimated from the outputs of the M sensors as

$$\hat{s}_i(m) = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}(m) = \sum_{j=1}^{N} w_{ij} x_j(m) \qquad i = 1, \ldots, M \tag{18.18}$$

where the vector $\boldsymbol{w}_i$ is the i^{th} row of the matrix $\boldsymbol{W}$ that demixes one element of the mixed sources. The ICA aims to find a demixing matrix $\boldsymbol{W}$ such that the elements of $\hat{\boldsymbol{s}}(m) = \boldsymbol{W}\boldsymbol{x}(m)$ are probabilistically independent and hence have a zero mutual information.

18.4.3 Basic Assumptions in Independent Component Analysis

For the source signals to be identifiable, via independent component analysis, it is required that the following assumptions are satisfied:

(1) The M source signals are statistically independent; this implies that their correlation matrix and their higher order cumulant matrices are diagonal. The most complete definition of independence is that the probability of a source vector is the product of the probabilities of the individual sources i.e. $f(\boldsymbol{s}) = f(s_1), f(s_2) \ldots f(s_M)$.

(2) Only one of the signal sources s_k may be Gaussian. A mix of two Gaussian signals cannot be separated as Gaussian processes are symmetric and do not possess higher than second order (i.e. covariance) matrix statistics. This issue is explained in detail later.

(3) The number of sensors N, must be at least as large as the number of estimated source components M i.e. the mixing matrix $\boldsymbol{A}$ must be of full rank.

18.4.4 The Limitations of Independent Component Analysis

(1) ICA cannot identify the scale or the sign of a mixing transformation since the transform $\boldsymbol{A}$ and the source signal $\boldsymbol{s}$ can be multiplied by a sign and a scaling factor and their reciprocals as

$$\boldsymbol{x}(m) = \boldsymbol{A}\boldsymbol{s}(m) = (-\alpha\boldsymbol{A})\left(-\frac{1}{\alpha}\boldsymbol{s}(m)\right) \tag{18.19}$$

(2) ICA cannot determine the position of the components in the source vector since any permutation of the source can be cancelled by an inverse permutation of the mixing matrix as

$$\boldsymbol{x}(m) = \boldsymbol{A}\boldsymbol{s}(m) = \boldsymbol{A}\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{s}(m) \tag{18.20}$$

where $\boldsymbol{P}$ is a permutation matrix.

(3) ICA cannot separate the mixed signals if more than one of the mixed signals is Gaussian, this is due to the symmetric distribution of Gaussian processes as explained next.

18.4.5 Why a mixture of two Gaussian signals cannot be separated?

A Gaussian process is a second order process; this means that it is completely defined by its mean vector (first order statistics) and covariance matrix (second order statistics) and all its higher than second order statistics are zero. Here we show that ICA cannot demix a mixture of two Gaussian processes.

Consider a combination of two independent Gaussian signal processes s_1 and s_2

$$x_1 = a_{11}s_1 + a_{12}s_2 \tag{18.21}$$

$$x_2 = a_{21}s_1 + a_{22}s_2 \tag{18.22}$$

Now consider the probability of the two independent signals s_1 and s_2

$$f(s_1, s_2) = f(s_1)f(s_2) = \frac{1}{2\pi}\exp\left(-0.5\left(s_1^2 + s_2^2\right)\right) \tag{18.23}$$

where for simplicity it is assumed that s_1 and s_2 have zero mean and unit variances. Since the joint probability distribution of s_1 and s_2 is symmetric and for equal probability pairs of values of s_1 and s_2 the argument of the exponential function in Equation 18.23 is the location of a circle, it follows that an infinite number of values of s_1 and s_2 on the circle can have the same value of $s_1^2 + s_2^2$ and hence the same probability. Now assuming that an estimate of $\boldsymbol{s} = [s_1,\ s_2]^{\mathrm{T}}$ is obtained as $\hat{\boldsymbol{s}} = \boldsymbol{Ax}$, we have

$$\begin{aligned} f(\hat{s}_1, \hat{s}_2) &= |\det\boldsymbol{A}|^{-1}\frac{1}{2\pi}\exp\left(-0.5\left(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{Ax}\right)\right) \\ &= |\det\boldsymbol{A}|^{-1}\frac{1}{2\pi}\exp\left(-0.5\left(\hat{s}_1^2 + \hat{s}_2^2\right)\right) \end{aligned} \tag{18.24}$$

Again, there are an infinite number of values of $\hat{s}_1$ and $\hat{s}_2$ that yield the same probability in Equation (18.24) and hence the same information measure.

18.4.6 The Difference Between Independent and Uncorrelated

ICA may not be achieved if the only optimisation criterion used is that the separated signals should be uncorrelated (in terms of second order statistics) and hence have a diagonal covariance matrix. This is because two signals may have a common component and at the same time may also have a zero cross-correlation and a diagonal covariance matrix. For a simple example $x_1(t) = \cos(2\pi ft)$ and $x_2(t) = \cos\left(2\pi ft + \frac{\pi}{2}\right) = \sin(2\pi ft)$ have the same frequency and amplitude but have a zero covariance despite the fact that one can be obtained from the other by a simple time shift. Hence the terms uncorrelated and independent have different meanings and implications.

Generally two variables s_{i} and s_{j} are uncorrelated, and have diagonal covariance matrix, if their covariance is zero

$$E[s_i s_j] - E[s_i]E[s_j] = 0 \tag{18.25}$$

For zero-mean random variables Equation (18.25) becomes $E[s_i s_j] = 0$.

Independence is a stronger property than uncorrelated; whereas two independent signals are uncorrelated, two uncorrelated signals are not necessarily independent. The condition for two signals to be independent is that any two functions of them need to have zero covariance, i.e.

$$E[h_1(s_i)h_2(s_j)] - E[h_1(s_i)]E[h_2(s_j)] = 0 \tag{18.26}$$

where $h_1(\cdot)$ and $h_2(\cdot)$ are any two functions; for example $h_1(s_i)h_2(s_j) = s_i^2 s_j^2$ etc.

To achieve ICA the signal vector is required to have independent components, in a probabilistic sense, such that the probability of the observation vector is the product of the probabilities of its components:

$$f(\boldsymbol{s}) = \prod_{i=1}^{M} f(s_i) \tag{18.27}$$

As explained in the next section, ICA can be achieved using entropy maximisation based on the idea that the entropy of independent components is greater than the entropy of correlated components. In the context of ICA, the maxent (maximum entropy) criterion is equivalent to minimising the cross entropy between the pdfs of the transformed signal $f(\hat{\boldsymbol{s}})$ and the desired pdf of the source signal with its independent components $f(\boldsymbol{s}) = \prod_{i=1}^{M} f(s_i)$. This cross entropy minimisation is often expressed in terms of optimisation of an objective function such as the diagonalisation of the higher order statistics and cumulants.

18.4.7 Independence Measures; Entropy and Mutual Information

The independence property of the signal components of a multi-variate vector may be quantified by the mutual information of the signals or a number of related measures, such as the non-Gaussianity measure, the maximum entropy (maxent) information measure, and diagonal cumulant matrices as discussed in this section. First, we introduce the important concepts of differential entropy and mutual information.

18.4.7.1 Differential Entropy

The entropy of an M-valued discrete random variable $Y \in [y_1, \ldots, y_M]$ is defined as

$$H(Y) = -\mathcal{E}(\log P_i) = -\sum_{i=1}^{M} P_i \log P_i \tag{18.28}$$

where P_i is the probability of the i^{th} symbol y_i and the logarithm is to the base of 2.

For a continuous-valued random process Y, the differential entropy information measure is defined as

$$H(Y) = -\mathcal{E}(\log f(y)) = \int_{-\infty}^{\infty} f(y) \log f(y) dy \tag{18.29}$$

where $f(y)$ is the probability density function of y and the logarithm is to the base of 2. Entropy is the information metric for discrete-valued processes, similarly, differential entropy is often employed as the information metric for continuous-valued processes. Note that entropy and differential entropy can also be interpreted as the expectation of the log likelihood of a process.

18.4.7.2 Maximum Value of Differential Entropy

The entropy of a discrete-valued random variable attains a maximum value for a uniformly distributed variable, whereas the entropy of a continuous-valued random variable, i.e. the differential entropy, is largest for *a Gaussian variable compared to other distributions with the same covariance matrix. The differential entropy of a variable with a variance of σ^2 is bounded by the inequality*

$$H(Y) \le \log\left(\sqrt{2\pi e\sigma}\right) \tag{18.30}$$

where the maximum value of $\log\left(\sqrt{2\pi e\sigma}\right)$ is attained for a Gaussian proess.

Example 18.1

Find the differential entropy of a uniformly distributed random variable. Assume a uniformly distributed random variable y has minimum and maximum limits of $[a,\ b]$ and hence a pdf of $f(y) = \dfrac{1}{b-a}$. The Differential entropy can be obtained by substituting for the uniform pdf in Equation (18.66) to obtain $H(y) - log(b - a)$. Since the variance of a uniform variable is given by $\sigma^2 = (b - a)^2/12$, the entropy of a uniform variable in terms of its variance becomes $H(y) = \log\left(\sqrt{12}\sigma\right)$ which is smaller than the maximum value of $log\left(\sqrt{2\pi e}\sigma\right)$ given by equation(18.30).

Example 18.2

Show that the differential entropy of two independent variables are additive and hence generalise the result. Consider a two dimensional vector

$$\begin{aligned}
H([y_1, y_2]) &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(y_1, y_2)\log f(y_1, y_2)dy_1dy_2 \\
&= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(y_1)f(y_2)(\log f(y_1) + \log f(y_2))dy_1dy_2 \\
&= -\int_{-\infty}^{\infty} f(y_1)\log f(y_1)\underbrace{\int_{-\infty}^{\infty} f(y_2)dy_2}_{1}dy_1 - \int_{-\infty}^{\infty} f(y_2)\log f(y_2)\underbrace{\int_{-\infty}^{\infty} f(y_1)dy_1}_{1}dy_2 \\
&= -\int_{-\infty}^{\infty} f(y_1)\log f(y_1)dy_1 - \int_{-\infty}^{\infty} f(y_2)\log f(y_2)dy_2 \\
&= H(y_1) + H(y_2) \qquad (18.31)
\end{aligned}$$

similarly for N independent variables $[y_1,\ y_2, \ldots,\ y_N]$ we have

$$H\left([y_1, y_2, \ldots, y_N]\right) = \sum_{k=1}^{N} H(y_k) \qquad (18.32)$$

18.4.7.3 Mutual Information

The mutual information of two or more random variables is a measure of their dependence (or independence). The mutual information of discrete-valued elements was defined in Section 6.4.2.

The mutual information (MI) of the continuous-valued elements, y_i, of an N-dimensional vector $\boldsymbol{y} = [y_1,\ y_2, \ldots,\ y_N]$ can be defined in terms of the Kullback–Leibler distance between the joint probability density function $f(\boldsymbol{y})$ and the marginal probability density function $\prod_{k=1}^{N} f(y_k)$ as

$$MI\big(\boldsymbol{y}=[y_1,y_2,\ldots,y_N]\big)=\int_{-\infty}^{\infty} f(\boldsymbol{y})\log\left(\frac{f(\boldsymbol{y})}{\prod_{k=1}^{N} f(y_k)}\right)d\boldsymbol{y} \tag{18.33}$$

Note that if $f(\boldsymbol{y})=\prod\limits_{k=1}^{N} f(y_k)$, then $\log\left(\frac{f(\boldsymbol{y})}{\prod_{k=1}^{N} f(y_k)}\right)=\log(1)=0$ and the mutual information of the elements of $\boldsymbol{y}=[y_1,\ y_2,\ \ldots,\ y_N]$ are zero.

The mutual information can be expressed in terms of the differential entropies as

$$\begin{aligned} MI\big(\boldsymbol{y}=[y_1,y_2,\ldots,y_N]\big) &= \int_{-\infty}^{\infty} f(\boldsymbol{y})\log\big(f(\boldsymbol{y})\big)d\boldsymbol{y}-\int_{-\infty}^{\infty} f(\boldsymbol{y})\log\left(\prod_{k=1}^{N} f(y_k)\right)d\boldsymbol{y} \\ &= \left(\sum_{k=1}^{N} H(y_k)\right)-H(\boldsymbol{y}) \end{aligned} \tag{18.34}$$

Note that for independent variables $H(\boldsymbol{y})=\sum\limits_{k=1}^{N} H(y_k)$ and hence the mutual information is zero. The limits of the mutual information of N variables is $(N-1)H(y)\geq MI([y_1,\ldots,y_N])\geq 0$ where the upper limit of $(N-1)H(y)$ occurs for the case of identical variables $y_1=y_2=\cdots=y_N=y$.

Example 18.3

Find the differential entropy and the mutual information of $\boldsymbol{y}=[y_1,y_2,\cdots,y_N]$; assuming that: (i) $y_1,y_2,\cdots,y_N$ are independent identically distributed (IID) variables, (ii) $y_1,y_2,\cdots,y_N$ are identical variables.

In the first case of IID variables $f(y_1,y_2,\ldots,y_N)=f(y_1)f(y_2)\ldots f(y_N)$ and the differential entropy is equal to $H(\boldsymbol{y})=\sum\limits_{k=1}^{N} H(y_k)$ and hence from Equation (18.34) the mutual information is $MI\big(\boldsymbol{y}=[y_1,$ $y_2,\cdots,y_N]\big)=\left(\sum\limits_{k=1}^{N} H(y_k)\right)-H(\boldsymbol{y})\ =0$.

In the second case, since $y_1=y_2\cdots=y_N=y$ we have $f(y_1,y_2\cdots,y_N)=f(y)$. *Hence* the differential entropy is equal to $H(\boldsymbol{y})=H(y)$ and using Equation (18.34) the mutual information is $MI(y_1,y_2\cdots,y_N)=NH(y)-H(y)=(N-1)H(y)$.

18.4.7.4 The Effect of a Linear Transformation on Mutual Information

Consider a linear transformation $\boldsymbol{y}$=$\boldsymbol{Wx}$. The probability density function (pdf) of the vector $\boldsymbol{y}$ in terms of that of the pdf of the vector $\boldsymbol{x}$ (see Section 3.8) is given by

$$f_Y(\boldsymbol{y}=\boldsymbol{Wx})=|\det(\boldsymbol{W})|^{-1}f_X(\boldsymbol{W}^{-1}\boldsymbol{y}) \tag{18.35}$$

Hence the log-likelihood of the probability of a transformation is

$$\log\big(f_Y(\boldsymbol{y}=\boldsymbol{Wx})\big)=\log\left(\left|\det\left(\boldsymbol{W}\right)\right|^{-1}\right)+\log\big(f_X\big(\boldsymbol{W}^{-1}\boldsymbol{y}\big)\big) \tag{18.36}$$

Hence, using Equations (18.35) and (18.36) it follows that the differential entropy of $\boldsymbol{y}$ can be expressed in terms of the differential entropy of $\boldsymbol{x}$ as

$$\begin{aligned}
H_Y(\boldsymbol{y}) &= -\int\limits_{-\infty}^{\infty} f_Y(\boldsymbol{y}) \log f_Y(\boldsymbol{y}) d\boldsymbol{y} \\
&= -\int\limits_{-\infty}^{\infty} |\det(\boldsymbol{W})|^{-1} f_X(\boldsymbol{W}^{-1}\boldsymbol{y}) \log\left(|\det(\boldsymbol{W})|^{-1} f_X(\boldsymbol{W}^{-1}\boldsymbol{y})\right) d\boldsymbol{y} \\
&= -|\det(\boldsymbol{W})|^{-1} |\det(\boldsymbol{W})| \left(-\log|\det(\boldsymbol{W})| \underbrace{\int\limits_{-\infty}^{\infty} f_X(\boldsymbol{X}) d\boldsymbol{x}}_{1} \right. \\
&\qquad \left. + \underbrace{\int\limits_{-\infty}^{\infty} f_X(\boldsymbol{x}) \log(f_X(\boldsymbol{x})) d\boldsymbol{x}}_{-H(x)} \right) \\
&= H(\boldsymbol{x}) + \log|\det(\boldsymbol{W})| \qquad (18.37)
\end{aligned}$$

Note that in the development of Equation (18.37) the relationship $d\boldsymbol{y} = |\det(\boldsymbol{W})| d\boldsymbol{x}$ has been used. Hence, the change in the differential entropy due to a linear transformation $\boldsymbol{W}$ *is* $\log|\det(\boldsymbol{W})|$. For example by multiplication of a random variable by a scalar factor w, its differential entropy changes by $log|w|$.

From Equation(18.34,37) the mutual information of $\boldsymbol{y}$ is given by

$$MI([y_1, y_2, \ldots, y_N]) = \sum_{k=1}^{N} H(y_k) - H(\boldsymbol{x}) - \log|\det(\boldsymbol{W})| \qquad (18.38)$$

Note when the matrix $\boldsymbol{W}$ yields independent components the mutual information $MI(\boldsymbol{y} = \boldsymbol{Wx}) = \boldsymbol{0}$ which implies that $\sum_{k=1}^{N} H(y_k) = H(\boldsymbol{x}) + log|\det(\boldsymbol{W})|$.

18.4.7.5 Non-Gaussianity as a Measure of Independence

From the central limit theorem we have that a linear combination of many independent random variables has a Gaussian distribution. A mix of even two non-Gaussian variables is more Gaussian than the original individual source variables. Hence it follows that a measure of independence of signals is their non-Gaussianity; demixed signals are more non-Gaussian than the mixed signals. For signal separation, or principal component/feature analysis, ICA use methods that maximise the non-Gaussian property of the transformed signals. Two main measures of non-Gaussianity used in ICA methods are negentropy and Kurtosis defined next.

18.4.7.6 Negentropy: A measure of Non-Gaussianity and Independence

The negentropy of a process is a measure of its non-Gaussianity. The negentropy measure is based on differential entropy.

Whereas the entropy of a discrete-valued variable attains a maximum value for a uniformly distributed variable, the entropy of a continuous-valued variable, the differential entropy, is largest for *a Gaussian variable compared to other distributions with the same covariance matrix.*

Hence differential entropy can be used to define a measure of non-Gaussianity of a process $\boldsymbol{y}$, defined as its Negentropy $J(\boldsymbol{y})$, as the difference between the differential entropy of a random process $\boldsymbol{y}$ and that of a Gaussian process $\boldsymbol{y}_{\mathrm{Gauss}}$ with the same covariance matrix, as

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{Gauss}) - H(\boldsymbol{y}) \tag{18.39}$$

Negentropy is a nonnegative variable that is zero for a Gaussian process. The advantage of using negentropy is that it is an information theoretic measure. However, the problem in using negentropy is its high computational complexity. Estimating negentropy using Equation(18.39) would require an estimate of the pdf of the process. Simpler approximations of negentropy will be discussed next.

18.4.7.7 Fourth Order Moments – Kurtosis

In general the cumulant of four zero-mean variables x_1, x_2, x_3 and x_4 is defined as

$$\begin{aligned} Q(x_1, x_2, x_3, x_4) = & E[x_1x_2x_3x_4] - E[x_1x_2]E[x_3x_4] - E[x_1x_3]E[x_2x_4] \\ & - E[x_1x_4]E[x_2x_3] \end{aligned} \tag{18.40}$$

The 4th order cumulant of a zero-mean random variable x is defined as

$$Q(x) = E[x^4] - 3(E[x^2])^2 \tag{18.41}$$

where $E[x^2] = \sigma_x^2$ is the variance of x.

Kurtosis is defined as the 4th order cumulant normalised by the square of the variance as

$$k(x) = \frac{E[x^4]}{(E[x^2])^2} - 3 \tag{18.42}$$

For a Gaussian process the fourth order moment $E[x^4] = 3(E[x^2])^2$, hence from Equation (18.70) the 4th order cumulant and the kurtosis of a Gaussian process are zero. Therefore kurtosis, like negentropy, can be used as a measure of non-Gaussianity. Super-Gaussian signals have positive kurtosis whereas non-Gaussian signals have negative kurtosis.

18.4.7.8 Kurtosis-based Contrast Functions – Approximations to Entropic Contrast

Consider two vector valued random variables $\boldsymbol{s}$ and $\boldsymbol{y}$ with probability densities $p(\boldsymbol{s})$ and $p(\boldsymbol{y})$. Assume $\boldsymbol{s}$ is the target variable and we wish $\boldsymbol{y}$ to have the same probability distribution as $\boldsymbol{s}$. In general the distance between the probability distributions of two random variables can be measured using the Kullback-Leibler criterion (also known as cross entropy) as

$$K(p_s|p_y) = \int_{-\infty}^{\infty} p_s(x) \log \frac{p_s(x)}{p_y(x)} dx \tag{18.43}$$

A probability distribution can be approximated in terms of the Edgeworth series: that is in terms of its cumulants. Using an Edgeworth expansion it can be shown that the distance between the probability distributions of two random variables can be approximated in terms of the distance between their second order and fourth order moments as

$$K_{24}(\boldsymbol{y}|\boldsymbol{s}) = \frac{1}{4}\sum_{ij}\left(R_{ij}^{y} - R_{ij}^{s}\right)^2 + \frac{1}{48}\sum_{ijkl}\left(Q_{ijkl}^{y} - Q_{ijkl}^{s}\right)^2 \tag{18.44}$$

where the first term on the r.h.s of Equation (18.44) is the distance between the covariances of the two processes R_{ij} and the second term is the distance between the fourth order cumulants of the two processes Q_{ij}.

Assuming that the source signals s_i are independent processes their covariance and cumulant matrices would be diagonal and hence we have

$$K_{24}(\boldsymbol{y}|\boldsymbol{s}) = \frac{1}{4}\sum_{ij}\left(R^{y}_{\boldsymbol{ij}} - \sigma^2(s_i)\delta_{ij}\right)^2 + \frac{1}{48}\sum_{ijkl}\left(Q^{y}_{ijkl} - k^2(s_i)\delta_{ijkl}\right)^2 \tag{18.45}$$

where $\sigma^2(s_i)$ and $k(s_i)$ are the variance and the kurtosis of the source signal $\mathrm{s_i}$ and the Kronecker delta function δ_{ijkl} is 1 when all its indices are identical, otherwise it is zero. Note that the minimisation of the distances in Equation(18.45) implies that ideally the diagonal elements of the covariances and cumulants of the estimate $\boldsymbol{y}$ are made identical to those of the source signal $\boldsymbol{s}$ whereas the off diagonal elements are minimised to zero.

In cases where a prior knowledge of the variance and cumulants of the source signals are not available, the cumulant contrast function can be simplified as

$$K_{24}(\boldsymbol{y}|\boldsymbol{s}) = \frac{1}{4}\sum_{ij\neq ii}\left(R^{y}_{ij}\right)^2 + \frac{1}{48}\sum_{ijkl\neq iiii}\left(Q^{y}_{ijkl}\right)^2 \tag{18.46}$$

Minimisation of the contrast function of Equation (18.46) implies the minimisation of the off-diagonal elements of the covariance and kurtosis functions.

18.4.8 Super-Gaussian and Sub-Gaussian Distributions

A Gaussian random process x has a 4th order moment related to its variance σ_x^2 as $E(x^4) = 3(E(x^2))^2 = 3\sigma_x^4$. Hence from Equation (18.42) the Kurtosis of a Gaussian process is zero. A zero mean uniform process of variance σ_x^2 has range of variations of $\pm\sqrt{12}\sigma/2$ and a probability of $f(x) = 1/(\sqrt{12}\sigma)$ and hence $\mathcal{E}(x^4) = \int_{-0.5\sqrt{12}\sigma}^{0.5\sqrt{12}\sigma} (1/(\sqrt{12}\sigma))x^4 dx = 1.8\sigma^4$. Hence from Equation (18.42) the kurtosis of a uniform process is -1.2.

Random processes with a positive kurtosis are called super-Gaussian and those with a negative kurtosis are called sub-Gaussian. Super-Gaussian processes have a spikier pdf than Gaussian, whereas subGaussian processes have a broader pdf. An example of a super Gaussian process is a Laplacian process.

Consider a general class of exponential probability densities given by

$$p(x) = k_1 x^{\beta} \exp\left(-k_2|x|^{\alpha}\right) \tag{18.47}$$

for a Gaussian signal $\alpha = 2$, $k_1 = 1/\sqrt{2\pi}\sigma$, $\beta = 0$ and $k_2 = 1/2\sigma^2$ where σ^2 is the variance. For $0 < \alpha < 2$ the distribution is super-Gaussian and for $\alpha > 2$ the distribution is sub-Gaussian.

Assuming that the signals to be separated are super-Gaussian, independent component analysis may be achieved by finding a transformation that maximises kurtosis. Figure 18.4 shows examples of Gaussian, subGaussian and subGaussian pdfs.

18.4.9 Fast-ICA Methods

Prior to application of a FastICA method, the data are sphered with PCA as explained in Section 18.3, so that the covariance matrix of the data is an identity matrix.

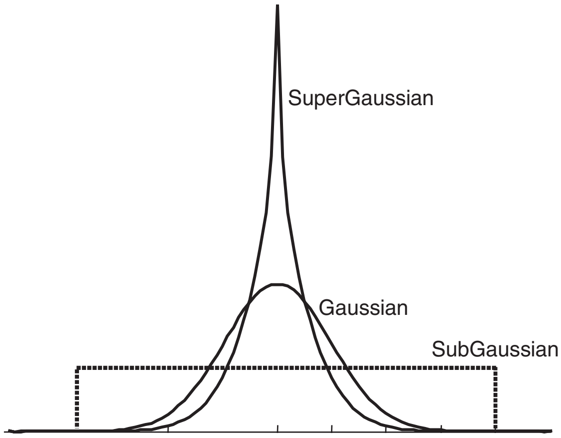


Figure 18.4 Super-Guassian pdfs are more peaky than Gaussian, whereas subGaussian pdfs are less peaky that Gaussian pdf.

The fast-ICA methods are the most popular methods of independent component analysis. Fast-ICA methods are iterative optimisation search methods for solving the ICA problem of finding a demixing matrix $\boldsymbol{W}$ that is the inverse of the mixing matrix $\boldsymbol{A}$ (within a scalar multiplier and an unknown permutation matrix).

These iterative search methods find a demixing transformation $\boldsymbol{W}$ that optimises a nonlinear contrast function. The optimisation methods are typically based on a gradient search or the Newton optimisation method that search for the optimal point of a contrast (objective) function $G(\boldsymbol{WX})$, where $\boldsymbol{X} = [x(0), x(1), \ldots, x(N-1)]$ is the sequence of observation vectors. At the optimal point of the contrast function the components of $\boldsymbol{y}(m) = \boldsymbol{Wx}(m)$ are expected to be independent.

18.4.9.1 Gradient search optimisation method

For gradient search optimisation the iterative update methods for estimation of the demixing matrix $\boldsymbol{W}$ is of the general form

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n + \mu g(\boldsymbol{W}_n\boldsymbol{X}) \tag{18.48}$$

where $g(\boldsymbol{W}_n\boldsymbol{X}) = \frac{d}{dW_n} G(\boldsymbol{W}_n\boldsymbol{X})$ is the first derivative of the contrast function and μ is an adaptation step size.

18.4.9.2 Newton optimisation method

For Newton optimisation the iterative methods for estimation of $\boldsymbol{W}$ is of the form

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n - \mu \frac{g(\boldsymbol{W}_n\boldsymbol{X})}{g'(\boldsymbol{W}_n\boldsymbol{X})} \tag{18.49}$$

where $g(\boldsymbol{W}_n\boldsymbol{X})$ is the first derivative of the contrast function and $g'(\boldsymbol{W}_n\boldsymbol{X})$ is the second derivative of the contrast function. The derivative of the contrast function, $g(\boldsymbol{W}_n\boldsymbol{X})$, is also known as the influence function.

18.4.10 Fixed-point Fast ICA

In the fixed-point FastICA a batch or block of data (consisting of a large number of samples) are used in each step of the estimation of the demixing matrix. Hence, each step is composed of the iterations on the samples that constitute the batch or block of data for that sample. In this section we consider the one-unit FastICA method where at each step one of the sources is estimated or demixed from the observation mixture, i.e. at each step one row vector $\boldsymbol{w}$ of the demixing matrix $\boldsymbol{W}$ is estimated.

A popular version of the fast ICA is based on a constrained optimization of the objective function G($\boldsymbol{wx}$) subject to the constraint $\mathcal{E}\big((\boldsymbol{w}^T\boldsymbol{x}(m))^2\big) = \|\boldsymbol{w}\|^2 = 1$. The solution is given by

$$\mathcal{E}\big(\boldsymbol{x}(m)\, g(\boldsymbol{w}^T\boldsymbol{x}(m))\big) - \beta \boldsymbol{w} = 0 \tag{18.50}$$

At the optimal value of $\boldsymbol{w}_0$, multiplying both sides of Equation (18.50) by $\boldsymbol{w}_0^T$ and noting that $\boldsymbol{w}_0^T\boldsymbol{w}_0 = \|\boldsymbol{w}_0\| = 1$ yields

$$\beta = \mathcal{E}\big(\boldsymbol{w}_0^T\boldsymbol{x}(m) g(\boldsymbol{w}_0^T\boldsymbol{x}(m))\big) \tag{18.51}$$

To obtain a Newton type solution, the Jacobian matrix of Equation (18.50) (the 2nd derivative w.r.t $\boldsymbol{w}$ in this case) is obtained as

$$JF(\boldsymbol{w}) = \mathcal{E}\big(\boldsymbol{x}(m)\boldsymbol{x}^T(m) g'(\boldsymbol{w}^T\boldsymbol{x}(m))\big) - \beta \mathbf{I} = 0 \tag{18.52}$$

where $F(\boldsymbol{w})$ denotes the left-hand-side of Equation (18.50) and $g'(\boldsymbol{w}^T\boldsymbol{x}(m))$ is the derivative of $g(\boldsymbol{w}^T\boldsymbol{x}(m))$ and the second derivative of $G(\boldsymbol{w}^T\boldsymbol{x}(m))$. Since it is assumed that prior to FastICA the data has been sphered so that $\mathcal{E}(\boldsymbol{x}(m)\boldsymbol{x}^T(m)) = \boldsymbol{I}$, the first term of Equation(18.52) can be approximated as

$$\mathcal{E}\big(\boldsymbol{x}(m)\boldsymbol{x}^T(m) g'(\boldsymbol{w}^T\boldsymbol{x}(m))\big) \approx \mathcal{E}\big(\boldsymbol{x}(m)\boldsymbol{x}^T(m)\big)\mathcal{E}\big(g'(\boldsymbol{w}^T\boldsymbol{x}(m))\big) = \mathcal{E}\big(g'(\boldsymbol{w}^T\boldsymbol{x}(m))\big) \tag{18.53}$$

Hence the Newton optimisation method, at the n^{th} iteration, can be written as

$$\begin{gathered} \boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \mu\left[\mathcal{E}\big(\boldsymbol{x} g(\boldsymbol{w}_n^T\boldsymbol{x}(m))\big) - \beta \boldsymbol{w}_n\right] / \left[\mathcal{E}\big(g'(\boldsymbol{w}_n^T\boldsymbol{x}(m))\big) - \beta \mathbf{I}\right] \\ \boldsymbol{w}_{n+1} = \boldsymbol{w}_{n+1} / \|\boldsymbol{w}_{n+1}\| \end{gathered} \tag{18.54}$$

where here $\boldsymbol{w}_{n+1}$ on the l.h.s represents the new value of the estimate at the n^{th} iteration.

18.4.11 Contrast Functions and Influence Functions

The nonlinear contract functions $G(\boldsymbol{Wx}(m))$ are the objective functions for optimisation of ICA transform $\boldsymbol{W}$: the optimal matrix $\boldsymbol{W}$ is a maxima (or minima) of $G(\boldsymbol{Wx}(m))$. The nonlinearity of the contrast function exploits the non-Gaussian distribution of the signal and facilitates decorrelation of the higher order statistics of the process after the second order statistics are decorrelated by a PCA pre-processing stage.

Optimisation of $G(\boldsymbol{Wx}(m))$ is achieved using either the gradient of the contrast function in an iterative optimisation search methods such as the gradient ascent (or descent) methods or the Newton optimisation method.

The gradient of the contrast function is given by

$$g(\boldsymbol{Wx}(m)) = \frac{\partial}{\partial W} G(\boldsymbol{Wx}(m)) \tag{18.55}$$

$g(\boldsymbol{Wx}(m))$ is also known as the influence function.

In general for a variable y with probability density function $p(y) \propto \exp(|y|^\alpha)$ the contrast function would be of the form

$$G(y) = [|y|^\alpha] \tag{18.56}$$

Hence for $\alpha = 2$, i.e. a Gaussian process $G(y) = E[|y|^2]$. For a super-Gaussian process $0 < \alpha < 2$. For sub-Gaussian processes α > 2. For highly super-Gaussian process when $\alpha < 1$, the contrast function shown in Equation (18.64) is not differentiable at $y = 0$ because in this case the differentiation of $\boldsymbol{y}$ would produce a ratio function with $\boldsymbol{y}$ at the denominator. Hence, for super-Gaussian functions, differentiable approximations to the contrast function are used. One such approximation is $log(cosh(y))$ whose derivative is $tanh(y)$. Another choice of nonlinearity for contrast function is $G(y) = -e^{-y^2/2}$ whose derivative is $g(y) = ye^{-y^2/2}$. For sub-Gaussian functions $G(y) = y^4$, Kurtosis, can be used as an appropriate choice of a contrast function.

Some of the most popular contrast and influence functions are as follows:

Contrast function	**Influence function**	**Appropriate process**
$G(y) = \log(\cosh(y))$	$g(y) = \tanh(y)$	General purpose
$G(y) = -e^{-y^2/2}$	$g(y) = ye^{-y^2/2}$	Highly super-Gaussian
$G(y) = y^4$	$g(y) = y^3$	Sub-Gaussian

Figure (18.5) illustrates three contrast functions and their respective influence functions. Note that the influence functions are similar to nonlinearities used in neural networks.

Note that it may be the case, perhaps it is often the case, that the process subjected to ICA is not identically distributed: that is different elements of the input vector process may have different forms of distributions, hence there is a case for using different contrast functions for different elements of the vector. This can be particularly useful for fine tuning stage of ICA, after application of a conventional ICA to the process. Of course one the needs to estimate the form of the distribution of each component and hence the appropriate contrast and influence functions.

18.4.12 ICA Based on Kurtosis Maximization – Projection Pursuit Gradient Ascent

The demixing of mixed signal may be achieved through iterative estimation of a demixing matrix $\boldsymbol{W}$ using an objective criterion that maximizes the kurtosis of the transformed signal vector. Let $\boldsymbol{x}(m)$ be the observation signal vector. In general each element of $\boldsymbol{x}(m)$ is a mixture of the source signals and hence $\boldsymbol{x}(m)$ has a non-diagonal covariance matrix. The signal $\boldsymbol{x}(m)$ is first decorrelated and diagonalised using the eigen vectors of the correlation matrix of $\boldsymbol{x}(m)$. Furthermore, the diagonalised process is sphered (i.e. made to have unit variance for each element of the vector process) by using a normalizing eigenvalue matrix. Let $\boldsymbol{z}(m)$ be the sphered (its covariance diagonalised and normalised) version of $\boldsymbol{x}(m)$

$$\boldsymbol{z}(m) = \boldsymbol{\Lambda}^{-0.5}\boldsymbol{U}^T\boldsymbol{x}(m) \tag{18.57}$$

where the matrices $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ are the eigen vectors and eigenvalues of the correlation matrix of $\boldsymbol{x}$.

As explained diagonalisation of the covariance matrix alone is not sufficient to achieve demixing. For demixing of a mixed observation we need to diagonalise the higher order cumulants of the signals. One way to achieve this is to search for a transform that maximises kurtosis which in the context of ICA is a measure of non-Gaussianity and independence.

Now assume that $\boldsymbol{w}_i$ is the i^{th} row vector of a demixing matrix $\boldsymbol{W}$ and that it demixes an element of $\boldsymbol{z}(m)$ as

$$y_i(m) = \boldsymbol{w}_i^T\boldsymbol{z}(m) \tag{18.58}$$

The fourth order of cumulant of $y_i(m)$ is given by

$$k(y_i(m)) = E(y_i^4(m)) - 3E(y_i^2(m)) = E\left((\boldsymbol{w}_i^T\boldsymbol{z}(m))^4\right) - 3E((\boldsymbol{w}_i^T\boldsymbol{z}(m))^2) \tag{18.59}$$

Assuming the signal is normalized, the instantaneous rate of change (differential) of kurtosis with $\boldsymbol{w}_i$ is given by

$$\frac{\partial k(y_i(m))}{\partial \boldsymbol{w}_i} = 4\boldsymbol{z}(m)\left(\boldsymbol{w}_i^T\boldsymbol{z}(m)\right)^3 \tag{18.60}$$

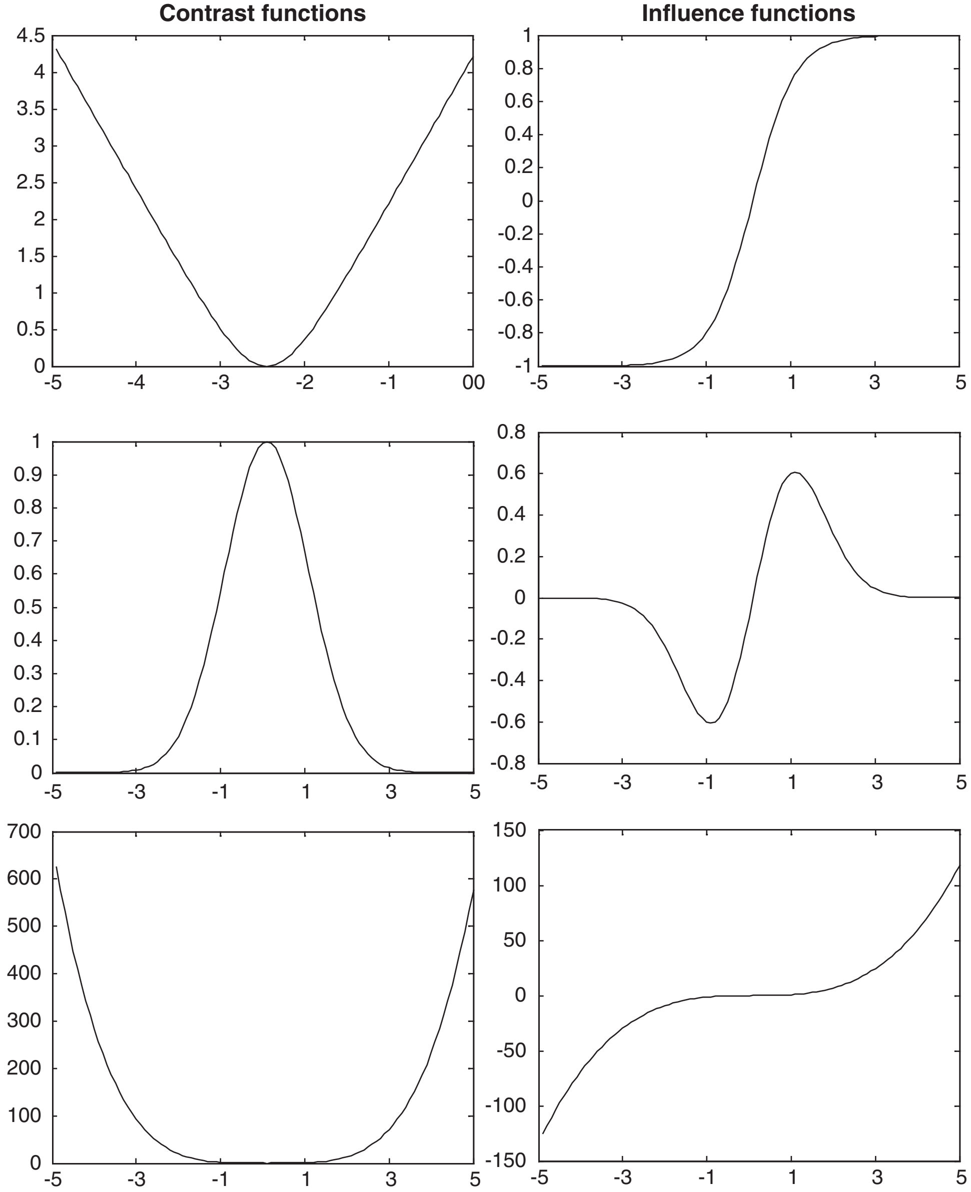


Figure 18.5 illustrations of three contrast functions and their respective influence functions, the x-axis represents the input to the influence/contrast functions.

An iterative gradient ascent identification method, for the demixing vector $\boldsymbol{w}_i$, based on kurtosis maximisation can be defined at iteration n as

$$\begin{aligned}\boldsymbol{w}_i(n) &= \boldsymbol{w}_i(n-1) + \mu \frac{\partial k(y_i(m))}{\partial \boldsymbol{w}_i(n-1)} \\ &= \boldsymbol{w}_i(n-1) + 4\mu \boldsymbol{z}(m)\left(\boldsymbol{w}_i^T \boldsymbol{z}(m)\right)^3 \qquad (18.61)\end{aligned}$$

where $\partial k(y_i(m))/\partial \boldsymbol{w}_i(n-1)$ is the rate of change of kurtosis with the transform coefficient vector. At the completion of each update $\boldsymbol{w}_i(n)$ is normalised as $\boldsymbol{w}_i(n)/|\boldsymbol{w}_i(n)|$.

Figure 18.6 illustrates a brain image and set of image basis functions obtained from ICA of the brain image. Contrasting these with the eigen-images of Figure 9.7 obtained from PCA method shows that ICA is more efficient as most of the information is packed into fewer independent subimages.

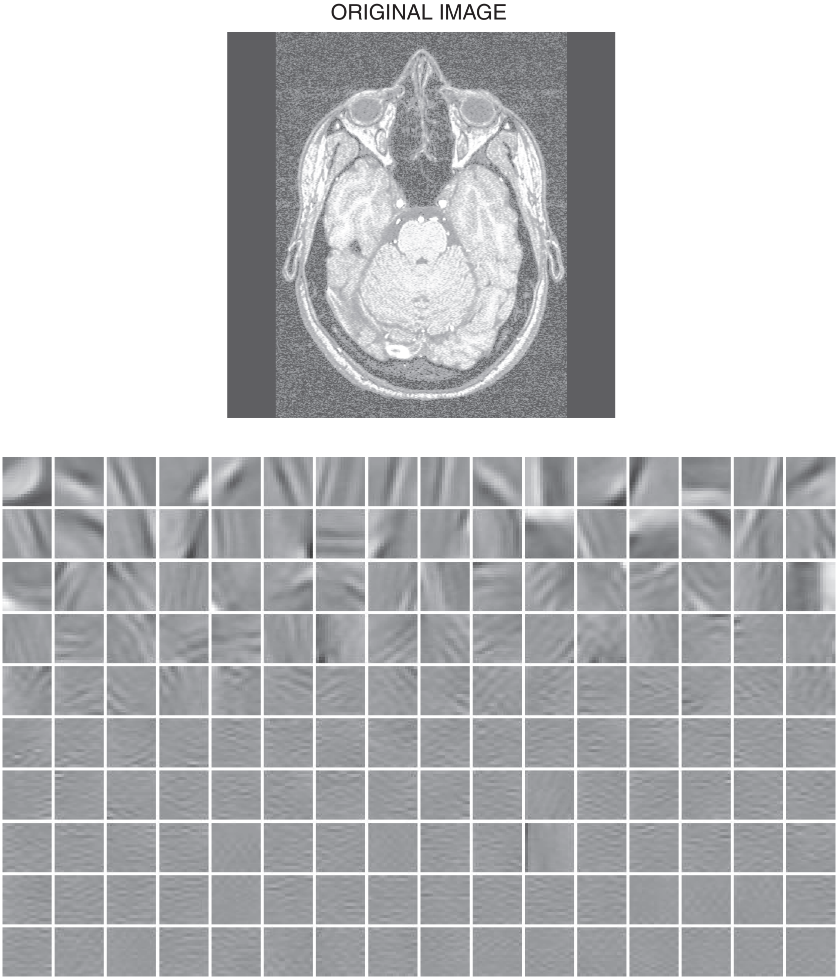


Figure 18.6 Top: a brain image, below: ICA based independent image basis functions obtained from the brain image of Figure 12.7. Contrast these with the eigen images of Figure 9.7 obtained from PCA method. Reproduced by permission of © 2008 Oldrich Vysata M.D., Neurocenter Caregroup, Rychnov nad Kneznou, Prague.

18.4.13 Jade Algorithm – Iterative Diagonalisation of Cumulant Matrices

The Jade algorithm (Cardoso and Saouloumiac) is an ICA method for identification of the demixing matrix. Jade method is based on a two stage process: (1) diagonalisation of the covariance matrix, this is the same as PCA and (2) diagonalisations of the kurtosis matrices of the observation vector sequence.

The Jade method is composed of the following stages:

(1) Initial PCA stage. At this stage first the covariance matrix of the signal $\boldsymbol{X}$ is formed. Eigen analysis of the covariance matrix yields a whitening (sphereing) matrix $\boldsymbol{W} = \boldsymbol{\Lambda}^{-0.5}\boldsymbol{U}^{\mathrm{T}}$, where the matrices $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ are composed of the eigen vectors and eigenvalues of the covariance of $\boldsymbol{X}$. The signal is sphered by transformation through the matrix $\boldsymbol{W}$ as $\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{X}$. Note that the covariance matrix of $\boldsymbol{Y}$ is $\mathcal{E}[\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}] = \mathcal{E}[\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}] = \boldsymbol{\Lambda}^{-0.5}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\,\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\,\boldsymbol{\Lambda}^{-0.5} = \boldsymbol{I}$.

(2) Calculation of kurtosis matrices. At this stage the fourth order (kurtosis) cumulant matrices $\boldsymbol{Q}_i$ of the signal are formed.

(3) Diagonalisation of kurtosis matrices. At this stage a single transformation matrix $\boldsymbol{V}$ is obtained such that all the cumulant matrices are as diagonal as possible. This is achieved by finding a matrix $\boldsymbol{V}$ that minimises the off-diagonal elements.

(4) Apply ICA for signal separation. At this stage a separating matrix is formed as $\boldsymbol{W}\boldsymbol{V}^{\mathrm{T}}$ and applied to the original signal.

As mentioned, the diagonalisation and sphereing of the covariance matrix of the observation process is performed using principal component analysis.

The diagonalisation of the cumulant matrices is a more complicated process compared to PCA mainly due to the four-dimensional tensorial nature of the 4th order cumulant matrices which have in the order of $O(N^4)$ parameters. Instead of using four dimensional matrices, the cumulant matrices are expressed in terms of a set of two-dimensional matrices with each element of the matrix expressed as

$$Q_{ij} = \sum_{k,l=1}^{M} cum(X_i, X_j, X_k, X_l) \tag{18.62}$$

Given T samples from each of the M sensors, a set of cumulant matrices can be computed as follows. Assume that the matrix $\boldsymbol{X}$ denotes an $M \times T$ matrix containing the samples from all the M sensors and that vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (the i^{th} and j^{th} row of $\boldsymbol{X}$) denote the T samples from sensors i and j respectively. The Jade algorithm calculates a series of $M \times M$ cumulant matrix $\boldsymbol{Q}_{ij}$ as

$$\boldsymbol{Q}_{ij} = (\boldsymbol{x}_i.\boldsymbol{x}_j.\boldsymbol{X})\boldsymbol{X}^T \qquad 1 \le i \le M, \quad 1 \le j < i \tag{18.63}$$

where $\boldsymbol{x}_i.\boldsymbol{x}_j$ is a vector dot product; that is an element by element multiplication of two $T \times 1$ vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The $T \times 1$ vector resulting for product of $\boldsymbol{x}_i.\boldsymbol{x}_j$ is then multiplied by every row vector of the $N \times T$ matrix $\boldsymbol{X}$. The result is then multiplied by $T \times M$ matrix $\boldsymbol{X}^{\mathrm{T}}$ to yield an $M \times M$ cumulant matrix $\boldsymbol{Q}_{ij}$.

The Givens rotation method is used to derive a matrix for the diagonalisation of the cumulant matrices. A Givens rotation matrix $\boldsymbol{R}(i, j, c, s)$ is equal to the identity matrix but for the following entries

$$\begin{pmatrix} \boldsymbol{r}_{ii} & \boldsymbol{r}_{ij} \\ \boldsymbol{r}_{ji} & \boldsymbol{r}_{jj} \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \tag{18.64}$$

where $c = \cos\theta$ and $s = \sin\theta$ are calculated to set the off diagonal elements to zero. For diagonalisation of the cumulant matrix, the angle θ, or equivalently the variables c and s, are calculated to minimise the following objective function over $K = \frac{M(M-1)}{2}$ cumulant matrices

$$O(c, s) = \sum_{k=1}^{K} OffDiag\big(\boldsymbol{R}(i, j, c, s)\boldsymbol{Q}_{ij}(k)\boldsymbol{R}^T(i, j, c, s)\big) \tag{18.65}$$

where OffDiag denotes the off diagonal elements of a matrix. It can be shown that the variables of c and s can be derived using the following method. Define a matrix $\boldsymbol{G}$ as

$$\boldsymbol{G} = \sum_{k=1}^{K} \left[Q_{ii} - Q_{jj} \; Q_{ij} + Q_{ji} \right]^{T} \left[Q_{ii} - Q_{jj} \; Q_{ij} + Q_{ji} \right] \tag{18.66}$$

Now define $x,\ y$ and r as

$$x = G(1,1) - G(2,2), \quad y = G(1,2) + G(2,1), \qquad r = \sqrt{x^2 + y^2} \tag{18.67}$$

The rotation angles for the ij off-diagonal elements are defined as

$$c = \sqrt{\frac{x+r}{2r}}, \quad s = \sqrt{\frac{y+r}{2r(x+r)}} \tag{18.68}$$

The above methods are described in Cardoso (1998) and Cardoso & Souloumiac (1996).

Figure 18.7 shows the application of Jade algorithm to separation of ECG of a mother and fetus where the measured signals from eight sensors have been used. The results show the efficiency and power of Jade method.

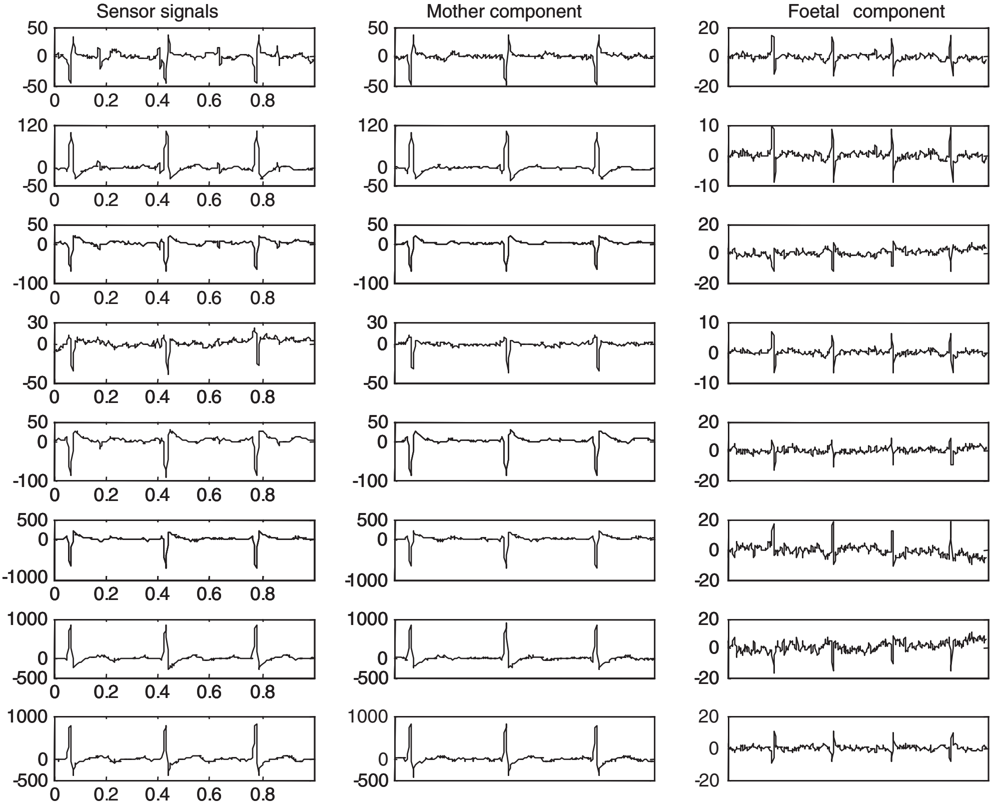


Figure 18.7 Application of Jade to separation of mother and fetus ECG. Note that signals from eight sensors are used in this example.

18.5 Summary

This chapter provided an introduction to MIMO systems and introduced a popular method, namely the independent component analysis (ICA) method, for demixing of MIMO signals. MIMO systems have a wide range of applications that include medical signal processing, mobile communication and microphone arrays. The various models for signal propagation in MIMO systems, namely instantaneous, anechoic and convolutional, were considered. Independent component analysis (ICA), introduced here as a signal demixing method, can be viewed as an extension of eigen analysis or PCA, however, there are substantial differences. Unlike PCA, ICA can be used to whiten and demix signals that are non-Gaussian. One important application of ICA is in blind source separation (BSS) for MIMO systems, another application is in principle feature extraction for non-Gaussian signals (such as eigen image extraction) as a more efficient alternative to PCA introduced in Chapter 9. The main feature of ICA is that it employs non-linear influence/contrast functions in the process of estimation of the optimal linear mixing transformation.

Bibliography

Benesty J., Chen J., and Huang Y., (2008) *Microphone Array Signal Processing*. Springer-Verlag, Germany.

Biglieri E., Calderbank R., Constantinides A., Goldsmith A. and Paulraj A, (2007) *MIMO Wireless Communications*, Cambridge University Press.

Golub G.H. and Loan C.F. Van (1996) *Matrix Computations*, 3rd edn, Johns Hopkins University Press.

Marcus, M. and Minc, H., (1988) *Introduction to Linear Algebra*. New York: Dover, p. 145.

Hyvärinen A., Karhunen J. and Oja E., (2001) *Independent Component Analysis*. John Wiley & Sons, Inc.

Bell A.J. and Sejnowski T.J. (1995) An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, **7**: 1129–1159.

Cardoso. J.F. (1999) Higher Order Contrasts for Independent Component Analysis, *Neural Computations*, **11**: 157–192.

Cardoso. J.F. (1998) Blined Source Separation: Statistical Principles, *Proc. IEEE*, 9(10): 2009–2025, Oct.

Cardoso J.F. (1997) Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, **4**:112–114.

Cardoso J.F., Souloumiac A., Jacobi (1996) Angles For Simultaneous Diagonalization *SIAM Journal on Matrix Analysis and Applications*.

Cichocki A., Bogner R.E., Moszczynski L., and Pope K (1997). Modified Herault-Jutten algorithms for blind separation of sources. *Digital Signal Processing*, **7**: 80–93.

Comon P. (1994) Independent component analysis–a new concept? *Signal Processing*, **36**: 287–314.

Cover T.M. and Thomas J.A. (1991), *Elements of Information Theory*. John Wiley & Sons, Inc.

Delfosse N. and Loubaton P. (1995) Adaptive blind separation of independent sources: a deflation approach. *Signal Processing*, 45:59–83.

The FastICA MATLAB package. http://www.cis.hut.fi/projects/ica/fastica/.

Friedman J.H. and Tukey J.W. (1974) A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Computers*, **23**(9): 881–890.

Friedman J.H. (1987) Exploratory projection pursuit. *J. of the American Statistical Association*, **82**(397): 249–266,.

Huber P.J. (1985) Projection pursuit. *The Annals of Statistics*, **13**(2): 435–475,.

Karhunen J., Oja E., Wang L., Vigário R., and Joutsensalo J. (1997) A class of neural networks for independent component analysis. *IEEE Trans. Neural Networks*, **8**(3): 486–504.

Lee T.-W., Girolami M., and Sejnowski T.J. (1999) Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, **11**(2): 417–441.

Van Trees H.L.Detection, (2002) *Estimation, and Modulation Theory, Part IV, Optimum Array Processing*. John Wiley & Sons, Inc.

Vigário R., Jousmäki V., Hämäläinen M., Hari R., and Oja E. (1998) Independent component analysis for identification of artifacts in magnetoencephalographic recordings. In *Advances in Neural Information Processing Systems 10*, MIT Press: 229–235.

Vigário R., Särelä J., and Oja E. (1998) Independent component analysis in wave decomposition of auditory evoked fields. *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*: 287–292, Skövde, Sweden.

19

Signal Processing in Mobile Communication

Mobile communication systems rely on digital signal processing methods for almost every essential function from locating and tracking the geographical positions of the mobile users to source coding (e.g. code excited linear prediction speech coders, MP3 music coders, JPEG image/video coders) to the division and allocation of time and bandwidth resources among the mobile users (e.g. time, space or code division multiple access systems).

Furthermore, the capacity and performance of mobile communication systems is limited not only by finite radio bandwidth but also by noise and interference. Hence the use of signal processing algorithms in noise reduction, channel equalisation, echo cancellation, error control coding, and more recently in space–time diversity signal processing using smart adaptive beam-forming antennas, are playing an increasingly pivotal role in the development of efficient and intelligent communication systems. In this chapter we consider how communication signal processing methods are used to improve the speed and capacity of communication systems.

19.1 Introduction to Cellular Communication

Mobile communication systems constitute one of the largest mass-market application areas of digital signal processing hardware and software systems. Whereas the early generations of mobile radio communication systems were relatively simple and bulky analogue radio devices connected to base stations, modern cellular communication systems are sophisticated digital signal processing systems to the extent that a modern mobile phone handset is a powerful special-purpose computer in its own right incorporating multimedia systems and functions in addition to a mobile phone. Current mobile phones integrate a host of multimedia services including video, speech, music, Internet and teleconferencing.

The increasing integration of multimedia services on mobile phones necessitates the use of broadband high-speed transmission systems for downloading and uploading data and that in turn requires new technology to develop better methods of utilisation of the available radio frequency spectrum. Bandwidth utilisation is an extremely important research and development area of mobile communications in which digital signal process theory and tools are essential. In the rest of this chapter we consider the applications of digital signal processing to mobile communication systems.

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi
© 2008 John Wiley & Sons, Ltd. ISBN: 978-0-470-75406-1

19.1.1 A Brief History of Radio Communication

The age of radio communication began in the 1860s with James Clark Maxwell's development of the theory of electromagnetic (EM) waves. Maxwell predicted the existence of electromagnetic radio waves with various frequencies propagating at the speed of light and concluded that light itself was also an electromagnetic wave. It appears that Maxwell did not realise that electromagnetic waves can travel in free space and assumed that some kind of 'ether' mediated the propagation of radio waves just as air mediates the propagation of sound (pressure) waves.

In 1884 Heinrich Rudolf Hertz reformulated Maxwell's equations. Later, between 1885 and 1888, in a series of pioneering experiments Hertz demonstrated that a rapidly oscillating electric current could be launched into space as an electromagnetic wave and detected by a wire loop receiver. For generating and transmitting oscillating radio waves Hertz used a high voltage induction coil and a capacitor (i.e. an LC oscillator circuit) connected to a rod with a gap in the middle and a spark sphere attached to the rod at each end of the gap, Figure 19.1. This device created oscillatory sparks across the gap as the spheres charged and discharged with opposite polarity electric charges. Like thunderstorms, sparks generate electromagnetic waves.

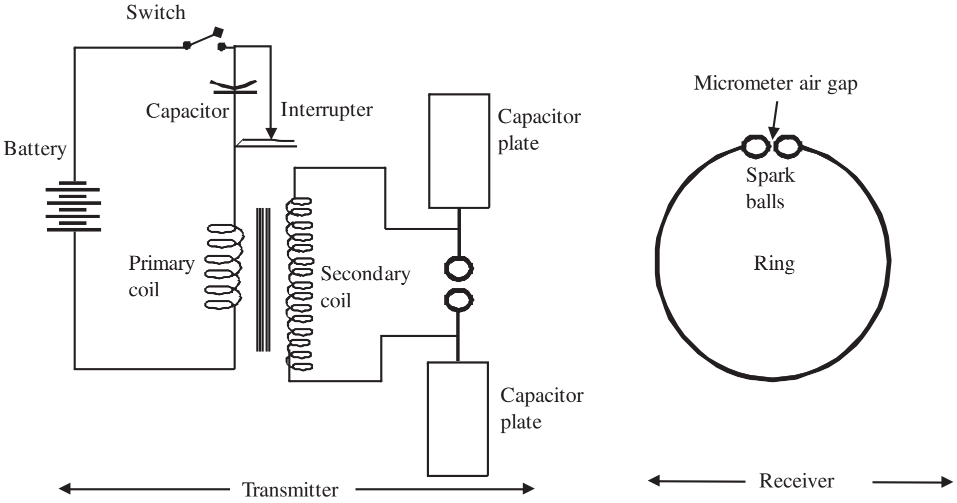


Figure 19.1 Schematic of Hertz's Experiment.

To detect the electromagnetic radiation, Hertz used a copper wire bent into a loop, with a small brass spheres connected to the ends with a very small gap between the spheres. The presence of an oscillating charge in the receiver wire loop caused sparks across the very small gap between the end points of the wire loop, Figure 19.1.

Hertz also demonstrated that radio waves had all the well-known properties of light waves – reflection from obstacles, diffraction from openings and around obstacles, refraction interference and polarisation. It is said that, in response to questions from his students who witnessed his classroom experiments on generation and reception of electromagnetic waves, Hertz replied that he saw no practical use for electromagnetic waves.

In 1895 Guglielmo Marconi, the inventor of radiotelegraph, inspired by Hertz's experiments, demonstrated the feasibility of radio communication by transmitting and receiving his first radio signal in Italy. In 1899 Marconi 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 the invention of radio communication. 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 of radio signals to noise and fading has been a major challenge in the design of wireless communication systems. Amplitude modulation (AM), where the radio carrier's amplitude is modulated by the signal massage, consumes no more bandwidth than the base band signal but is susceptible to noise and fading. High-quality radio transmission was made possible by Edwin Howard Armstrong's invention of wideband frequency modulation (FM) in 1933. In the FM modulation method, before transmission, amplitude variations of the signal are converted to frequency variations. FM can trade off more bandwidth for more robustness to noise and provides transparent broadcast quality at the cost of using more bandwidth than AM. FM transmission remained the main method for high-quality civilian radio transmission until the recent deployment of digital radio communication.

The era of mobile cellular communication began in the 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. The first generation of mobile phone systems had low user capacity and the large 'brick-like' handsets were cumbersome and power inefficient. 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 System for Mobile (GSM) communication 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 the third generation (3G) and universal mobile telecommunications system (UMTS) standards.

19.1.2 Cellular Mobile Phone Concept

The main limited resources in modern mobile communication systems are bandwidth and on-board battery power. The available radio bandwidth is limited by the need to share the finite radio frequency spectrum among many different mobile users and for many other different applications and purposes. The available power is limited by the capacity, size and weight of the onboard (handset) batteries. In a mobile communication system the radio frequency bandwidth needs to be used efficiently in order to maximise the capacity.

A major solution to the problem of limited radio channel capacity, resulting from finite bandwidth, is cellular systems. In a cellular system the same band of frequencies in suitably distant cells are reused to transmit different data. Figure 19.2 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.

Note that smaller cell size also requires less power as within smaller cells the base station and the mobile devices operate on less power. This is advantageous in terms of the onboard power requirements and also in terms of the possible electromagnetic radiation hazard.

A key aspect of cellular mobile communication technology is the very large increase in spectral efficiency achieved through the arrangements of cells in clusters and the *reusing* of the same radio frequency channels in non-adjacent cells; this is possible because in a cellular systems the cell phones and base stations operate on low-power transmitters whose electromagnetic wave energy fades away before they reach non-adjacent cells. Note that a cellular system also results in a more efficient use of batteries due to less power used in smaller cells.

In the cellular configuration of Figure 19.2(a), each cell, within each cluster of seven cells, uses a set of different frequencies (each set of distinct radio channel frequencies used in a cell is shown by a different number). Different clusters of seven cells reuse the same sets of frequencies as shown by the number codes in each cluster of cells.

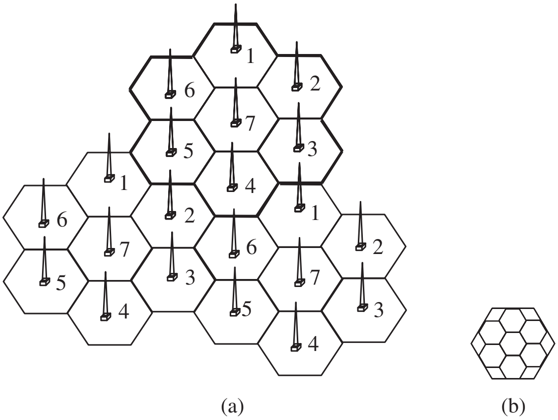


Figure 19.2 (a) A cellular system increases capacity by dividing a geographical area into clusters of cells and reusing the radio spectrum in non-adjacent cells. Each set of radio channel frequencies used in a cell is shown by a different number. (b) In an area with large user demand, a cell can be split into several subcells to further increase the overall capacity of the mobile communication system.

The system capacity of a cellular phone system can be increased by reducing the cell size and transmission power. The cost is of course more cell infrastructure. In busy population centres, such as city centres, where there is a much higher than average demand for network service, smaller cells can be used by a process of cell-splitting. Figure 19.2(b) shows the process of cell splitting which is used in high density population areas with high user demand in order to increase the overall channel capacity of the cellular systems.

19.1.3 Outline of a Cellular Communication System

Figure 19.3 shows the outline of basic network architecture of a GSM cellular mobile phone system. Each cell has a base station that accommodates the transmitter/receiver antennas, the switching networks and the call routing and all the call management equipment of the base station. The basic cellular system consists of the following subsystems.

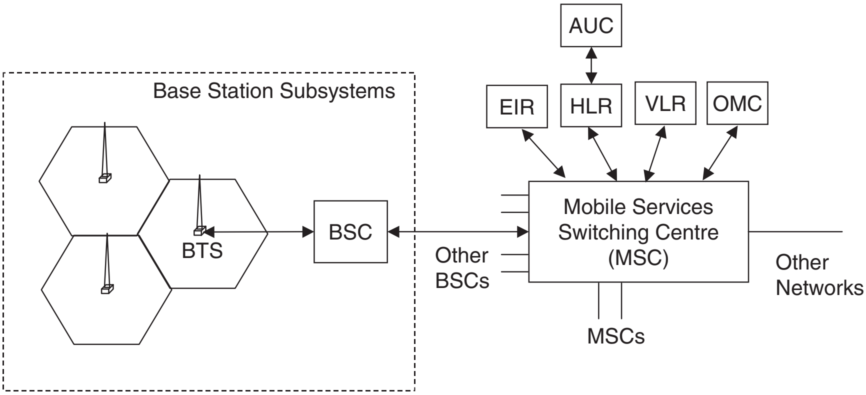


Figure 19.3 Illustration of a basic GSM mobile phone network.

(1) **MS** – Mobile Station. The MS is the physical equipment used by a subscriber; usually this is a hand-held cellular telephone handset. It is comprised of two distinct elements, the ME (Mobile Equipment) and the SIM (Subscriber Identity Module). The ME equipment consists of the screen, speaker, microphone, keypad, antenna and all the inside hardware and software elements including signal processing microchips. The SIM contains information that enables the user to access the mobile network.

(2) **SIM** – Subscriber Identity Module is a smart card which stores the key number identifying a mobile phone service subscriber, as well as subscription information, preferences, network state information such as its current location area identity (LAI) and text messages. If the handset is turned off and back on again it will take data off the SIM and search for the LAI it was in; this avoids searching the whole list of frequencies that the telephone normally would. Each SIM is uniquely identified by its International Circuit Card ID (ICCID).

(3) **BTS** – Base Transceiver Station, usually placed on top of tall buildings or on poles, contains the equipment for transmission and reception of radio signals (transceivers), antennas, and equipment for encrypting and decrypting communication with the Base Station Controller (BSC). Typically a BTS will have several transceivers which allow it to serve several different frequencies and different sectors of the cell in the case of sectorised base stations. A typical BTS site may have from 1 to 12 transceivers in one, two or three sectors, although these numbers may vary widely.

(4) **BSC** – Base Station Controller controls the BTSs and is responsible for allocation and release 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. Typically a BSC has 10s or 100s of BTSs under its control. The databases for all the sites, including such information as carrier frequencies, frequency hopping lists, power reduction levels, receiving levels for cell border calculation, are stored in the BSC. This data is obtained directly from radio planning engineering which involves modelling of the signal propagation as well as traffic projections. A main function of the BSC is to act as a concentrator where many different low-capacity connections to BTSs are reduced to a smaller number of connections, with a high level of utilisation, towards the Mobile Switching Center (MSC).

(5) **MSC** – Mobile Switching Centre is responsible for switching and routing of telephone calls. MSC provides circuit-switched calling, mobility management, and services to the mobile phones roaming within the area that it serves. This means voice, data and fax services, as well as SMS and call divert. The tasks of the MSC include delivering calls to subscribers as they arrive in a cell based on information from the VLR (Visitor Location Register); connecting outgoing calls to other mobile subscribers or the PSTN; delivering SMSs from subscribers to the SMS centre (SMSC) and vice versa; arranging handovers between BSCs; carrying out *handovers* from and to other MSCs and supporting supplementary services such as conference calls, call hold and billing information.

Gateway MSC is the MSC that determines the location of the visited MSC where the subscriber who is being called is currently located. It also interfaces with the Public Switched Telephone Network. All mobile-to-mobile calls and PSTN-to-mobile calls are routed through a GMSC. The **Visited MSC** is the MSC where a subscriber is currently located. The VLR associated with this MSC will have the subscriber's data in it. The **Anchor MSC** is the MSC from which a handover has been initiated. The **Target MSC** is the MSC towards which a handover should take place.

The MSC connects to the following elements: the home location register (HLR) for obtaining data about the SIM and MSISDN; the Base Station Subsystem which handles the radio communication with mobile phones; the network which handles the radio communication with mobile phones; the VLR for determining where other mobile subscribers are located.

(6) **HLR** – Home Location Register is a central database that holds general information about mobile phone subscribers. The main function of the HLR is to track and manage the movements and location area of mobile SIMs and phones. HRL implements the following procedures: manages and tracks the mobility of subscribers by updating their position in administrative areas called 'location

areas'; sends the subscriber data to a VLR when a subscriber first roams there; brokers between the mobile switching centre and the subscriber's current VLR in order to allow incoming calls or text messages to be delivered and removes subscriber data from the previous VLR when a subscriber has roamed away.

The HLR database stores details of every SIM card issued by the mobile phone operator. Each SIM has a unique identifier which is one of the primary keys to each HLR record. The SIMs have associated telephone numbers used to make and receive calls to the mobile phone. Other data stored in the HLR in a SIM record is: GSM services that the subscriber has requested or been given, General Packet Radio Service (GPRS) settings to allow the subscriber to access packet services, current location of subscriber and call divert settings.

The HLR connects to the following elements: the Gateway MSC (GMSC) for handling incoming calls; the VLR for handling requests from mobile phones to attach to the network; the SMSC for handling incoming SMS; the voice mail system for delivering notifications to the mobile phone that a message is waiting.

(7) **VLR** – Visitor Location Register is a temporary database that holds visiting subscribers' general information during their visit to a cell. It also holds location area identities of roaming users. Each base station is served by one VLR.

The primary functions of the VLR are: to inform the HLR that a subscriber has arrived in the particular area covered by the VLR; to track where the subscriber is within the VLR area when no call is ongoing; to allow or disallow which services the subscriber may use; to allocate roaming numbers during the processing of incoming calls; to remove the subscriber record if a subscriber becomes inactive whilst in the area of a VLR. The VLR deletes the subscriber's data after a fixed time period of inactivity and informs the HLR (e.g. when the phone has been switched off and left off or when the subscriber has moved to an area with no coverage for a long time).

The data stored in the VLR has either been received from the HLR, or collected from the MS. Data stored in the VLR includes: the subscriber's identity number; authentication data; the subscriber's phone number; the services that the subscriber is allowed to access; access point subscribed and the HLR address of the subscriber.

The VLR connects to the following elements: the visited MSC (VMSC) to pass data needed by the VMSC during its procedures, e.g. authentication or call setup; the HLR to request data for mobile phones attached to its serving area; other VLRs to transfer temporary data concerning the mobile when they roam into new VLR areas.

(8) **EIR** – Equipment and Identity Register is a database that holds user equipment identities. The EIR is often integrated to the HLR. The EIR keeps a list of mobile phones which are to be banned from the network or monitored. This is designed to allow tracking of stolen mobile phones. The EIR data does not have to change in real time, which means that this function can be less distributed than the function of the HLR.

(9) **AUC** – When a phone is powered on, the authentication centre (AUC) authenticates the SIM card that attempts to connect to the network. Once the authentication is successful, the HLR is allowed to manage the SIM and services described above. An encryption key is also generated that is subsequently used to encrypt all wireless communications between the mobile phone and the core network. If the authentication fails, then no services is available to the mobile phone. The AUC does not engage directly in the authentication process, but instead generates data known as triplets for the MSC to use during the procedure. The security of the process depends upon a shared secret key, called the Ki, between the AUC and the SIM. The 128-bit secret key Ki is securely burned into the SIM during manufacture and is also securely replicated onto the AUC. This Ki is not transmitted between the AUC and SIM, but is combined with the International Mobile Subscriber Identity number (IMSI) to produce a challenge/response for identification purposes and an encryption key called Kc for use in over-the-air communications.

(10) **OMC** – Operation and Maintenance Centre is a database that holds relevant information about the overall operation and maintenance of the network. The OMC is connected to all equipment in the switching system and to the BSC. The implementation of OMC is called the operation and support system (OSS). The OSS is the functional entity from which the network operator monitors and controls the system. The purpose of OSS is to offer the customer cost-effective support for centralised, regional, and local operational and maintenance activities that are required for a GSM network. An important function of OSS is to provide a network overview and support the maintenance activities of different operation and maintenance organisations.

(11) **GMSC** – Gateway Mobile Switching Centre is the point to which a MS terminating call is initially routed, without any knowledge of the MS's location. The GMSC is thus in charge of obtaining the MSRN (Mobile Station Roaming Number) from the HLR based on the MSISDN (Mobile Station ISDN Number, the 'directory number' of an MS) and routing the call to the correct visited MSC.

(12) **SMS-G** – This is the term used to collectively describe the two Short Message Services Gateways described in the GSM recommendations. The SMS-GMSC (Short Message Service Gateway Mobile Switching Centre) is for mobile terminating short messages, and SMS-IWMSC (Short Message Service Inter-Working Mobile Switching Centre) for mobile originating short messages.

19.2 Communication Signal Processing in Mobile Systems

Modern mobile communication systems rely on advanced signal processing methods for fast, efficient, reliable and low-cost multimedia communication. In fact almost every aspect of the functioning of transmission and reception of information on a mobile phone is dependent on advanced signal processing systems. The signal processing functions in a mobile communication system includes the following modules:

(1) **Source Coder/Decoder** – At the transmitters source coders compress the number of bits per sample of the input signal (such as speech, image and music) by removing the correlation and redundancies from the signals; source decoders decompress and reconstruct the signals at the receiver. Source coding achieves a reduction in bit rate and hence a proportional reduction in the bandwidth and the power required to transmit the bits.

Source coding involves the modelling and utilisation of the correlation structure of the signal. This can be achieved through a combination of (i) the use of signal transforms such as discrete Fourier transform and cosine transform, (ii) the use of signal generation models such as linear prediction models and (iii) 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, and JPEG image coders can significantly reduce (often by a factor of more than 10 to 20) the required bit rate and bandwidth and hence increase the capacity and speed of transmission of audio, image, text and other data. Speech and music compression are covered in Chapters 13 and 14.

(2) **Channel Coder/Decoder** – The purpose of channel coding is to introduce the ability to detect and correct bit errors and hence reduce transmission errors due to noise, fading and loss of data packets. The simplest (and not very efficient) form of a channel coder is a repetition code, for example instead of transmitting the bits '1' or '0' the bit pairs '11' or '00' are transmitted.

In mobile phones channel coding involves the use of a combination of convolution and block coders for the addition of error-control bits to the source data in order to increase the distance between the allowed sequences of transmitted data and hence improve the error detection and error correction capability of communication systems.

(3) **Multiple Access Signalling** – In a communication system time and bandwidth needs to be shared between different users. Multiple access signalling, as the name implies, provides simultaneous access

to multiple users on the same shared bandwidth and time resources. Multiple access systems are based on division of time or 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. In code division multiplexing, orthogonal codes are assigned to different users who can then communicate simultaneously on the same bandwidth.

(4) **Cell Handover** – The tracing and determination of the geographical location of a mobile phone user and the issue of which cell (and antenna) at any given time should serve a mobile user, as the user roams across different cells, is accomplished by processing the strengths of the radio signals from a mobile user received by different base stations. Usually the base station, or a number of base stations, that receives 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 metres or less depending on the cell size.

(5) **Rake Correlators** – In a broadband mobile environment the multi-path reflections of an electromagnetic signal, from different surfaces and through different paths, arrive at different times as several distorted replicas of the transmitted signal. Rake receivers advantageously use the effect of the multi-path propagation by combining the reflections of a signal received from different propagation paths. This form of the so-called space–time diversity signal processing can reduce fading and add to the strength of the received signal.

(6) **Channel Equalisation** – This is used to remove the distortions and time-dispersion of signals that result from the non-ideal characteristics of radio channels. Channel equalisation requires the identification of the channel response and the implementation of the inverse channel response i.e. the equaliser. Channel equalisation reduces the symbol overlaps (ISI) and bit error rate at the receiver. Channel equalisation is described in Chapter 17.

(7) **Echo Cancellation** – This is used to reduce both acoustic feedback echo between the speaker and the microphone of a mobile phone and also the telephone exchange hybrid line echo. Echo cancellation is necessary for voice and data communication systems. Acoustic echo cancellation is particularly important for mobile phones, hands-free phones and teleconferencing systems. Echo cancellation is described in detail in Chapter 16.

(8) **Smart Antenna Array** is 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 is controlled by adaptive signal processing methods so that the transmitted electromagnetic power is more efficiently beamed and selectively directed towards the mobile users.

19.3 Capacity, Noise, and Spectral Efficiency

A principal challenge in the design and development of mobile communication systems stems from the limitations imposed on data transmission rates due to finite bandwidth and the physical properties of the radio communication channels, which can include noise and distortions such as signal dispersion, fading, impulsive noise, co-channel/multiple-access interference as well as other phenomena most notably multi-path effects.

The rate at which binary data bits (i.e. sequences of ones and zeros) can be transmitted on a communication channel is limited by (a) the available bandwidth and (b) disturbances such as noise, interference, distortion and multi-path effects.

Binary data are signalled using communication symbols. In its simplest form a communication symbol may be a pulse-modulated sinusoidal carrier 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 simultaneously carry M bits provided that there are 2^M resolvable patterns of variation of the symbol such as the amplitude, or phase or time delay (position) or frequency of the symbol. Assuming each symbol carries M bits, the bit rate is M times the symbol rate, i.e. $r_b = M\ r_s$. 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 communication symbol can carry in an M-ary signalling scheme is limited by noise, interference, multi-path 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 multi-level encoding techniques are used. This is because at low signal-to-noise ratios (SNR) the noise can obliterate the 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 it follows that 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

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \tag{19.1}$$

where C is the channel capacity in bits per second; 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 versus the levels of noise and interference. Equation (19.1) gives the theoretical limit that the best possible coding and modulation method may achieve. Figure 19.4 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 per Hz, at 15 dB it is 5 bits per Hz and this approaches a theoretical rate of 13 bits per Hz at 40 dB SNR.

The challenge in communication signal processing is to increase the capacity in bits per second per Hz of bandwidth through reducing noise, interference and multi-path distortion. Signal processing methods play a central role in removing or compensating for the effect of noise and thereby improving data transmission capacity.

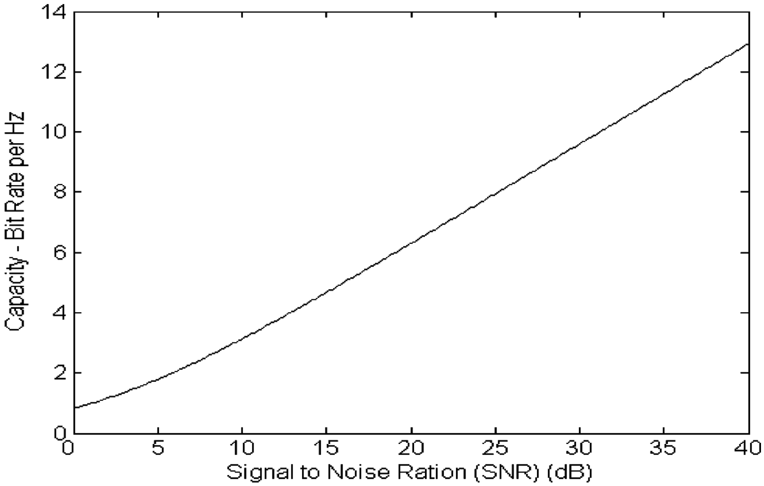


Figure 19.4 Illustration of the variation of the maximum capacity (bits/sec/Hz) of a communication channel with the SNR (dB).

The signal processing methods that are used to improve the capacity of mobile communication systems include source coding, channel coding, channel equalisation, echo cancellation, multi-path 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 is opening a myriad of possibilities in the use of signal processing for enhanced mobile communication.

19.3.1 Spectral Efficiency in Mobile Communication 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 mega Hz, usually centred between 1 to 4 GHz. The very large number of subscribers using mobile communication devices and the ever-increasing demand on the bandwidth is accommodated through the efficient use of communication resources that results in a large increase in the capacity per Hz, also known as the spectral efficiency defined as the data rate in bits per second per Hz unit of bandwidth:

$$\text{spectral efficiency} = \text{channel throughput/channel bandwidth}$$

Depending on the efficiency of the communication systems and the signal-to-noise ratio, the actual spectral efficiency may vary from 0.1 to 4 bps/Hz.

The main method currently used to increase the capacity of radio channels is based on frequency reuse. As explained earlier this involves the reusing of the same frequencies in non-adjacent cells where the power of the transmitted electromagnetic wave from a cell fades to insignificance by the time it reaches the non-adjacent 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, as shown in Figure 19.2(b). In order to minimise the interference among non-adjacent 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: Due to the low power of the transmissions from base-stations and mobile phones, the electromagnetic waves from each cell fades away before they reach non-adjacent 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 cell phones is relatively low. Low power consumption implies small handset batteries with longer talk time and also less exposure to possibly harmful electromagnetic radiation.

19.4 Multi-path and Fading in Mobile Communication

In classical communication theory it is assumed that the received signal is corrupted by additive white Gaussian noise (AWGN) and the signal distortion due to the channel response is modelled by inter-symbol interference. In reality there are many different sources of noise and interference that may limit the performance of a communication 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 multi-path and fading, described next.

19.4.1 Multi-path Propagation of Electromagnetic Signals

Three main mechanisms impact the propagation of a radio frequency (RF) electromagnetic wave in a mobile communication environment. These are:

(1) *Reflection* occurs when an electromagnetic wave impinges a smooth surface with much larger dimensions than the wavelength λ of the radio frequency (RF) electromagnetic signal. Note that 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. In comparison light has a wavelength of 700 nm for red to 400 nm for violet.
(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 several 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 from rough objects with large dimensions or from any object with dimensions comparable or smaller that the wavelength λ of the RF electromagnetic signal, causing the wave to scatter in all directions.

In a wireless communication environment the transmitted electromagnetic wave, usually encounters a number of different obstacles, reflectors and diffractors 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 called the *multi-path effect*. For mobile systems the communication environments change with time, space and the speed of movement of the mobile user and these result in time-varying multi-path channel effects.

A simple illustration of multi-path effects in wireless connections is shown in Figure 19.5. The multi-path effect is usually described by:

(1) Line-of-sight path: this is the direct path between the transmitter and the receiver antennas.
(2) None-line-of-sight paths: the paths arriving after refraction and reflections from various objects and surfaces.

A multi-path propagation effect can be modelled as the impulse response of a linear channel as

$$h(t) = \sum_{l=1}^{L} \alpha_l \delta(t - \tau_l) \tag{19.2}$$

where L is the number of different propagation paths. The multi-path Equation (19.2) has two parameters for each propagation path l; these are the propagation time delay τ_l and the amplitude fading factor α_l.

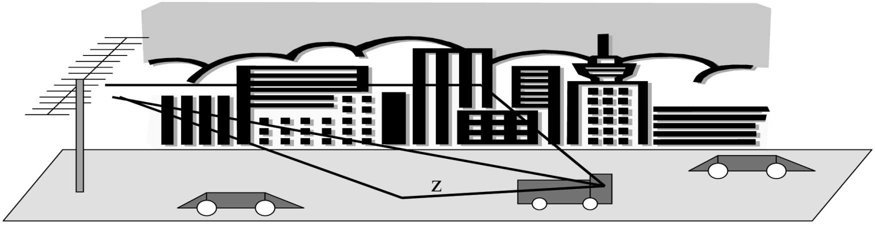


Figure 19.5 A simple illustration of multi-path propagation effects in a mobile phone environment.

The multi-path effect results in amplitude and phase fluctuations and time delay in the received signals and this can reduce the transmission capacity and in severe cases create signal outage and loss of connection. The multi-path effect is described by the following characteristics:

(1) *Multi-path fading characteristics*: When the reflected waves, which arrive 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 which occurs due to movements of a mobile user over large areas and fast fading which occurs due to movements over smaller distances comparable to the wavelength.
(2) *Multi-path 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 characteristic of the channel is called multi-path delay spread, which is an important parameter for assessment of the performance of wireless systems.

A common measure of multi-path delay spread is the root mean square (rms) delay spread. For reliable communication without using adaptive equalisation or other multi-path 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 a flat channel or narrowband channel. When the transmitted data is equal to or larger than the coherent bandwidth, the wireless channel is called a frequency-selective channel or wideband channel.

19.4.2 Rake Receivers for Multi-path Signals

The actual form of the signal distortion due to multi-path reflections depends on the signal bandwidth, the durations of transmitted and reflected signals and on the time delays incurred in propagation of a signal via different reflection paths.

The duration of a signal is inversely proportional to its bandwidth. For very wideband very short duration signals, multi-path reflection does not result in inter-symbol interference, rather it results in the appearance of multiple distorted replicas of the transmitted pulses. For example at a bandwidth of 10 megaHz the duration of a pulse is roughly about 0.1 microseconds. Now the distance travelled by an electromagnetic wave in 0.1 microsecond is only 30 metres. Hence, any two reflections of pulses travelling distances with differences of more than 30 metres 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 multi-path components of a signal. The correlators' outputs are combined to achieve improved SNR and communications reliability and performance. This is a type of the 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. Two primary methods are used to combine Rake-receiver finger outputs. One method weights each output equally. The second method uses the data to estimate weights which maximise the SNR of the combined output.

19.4.3 Signal Fading in Mobile Communication Systems

Modelling the fading and the propagation loss of electromagnetic wave energy in space is important for the calculation of the required transmitter power in mobile communication systems. In the idealised 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 a free space with the square of the distance d as

$$L_s(d) = \left(\frac{4\pi d}{\lambda}\right)^2 = \left(\frac{4\pi df}{c}\right)^2 \tag{19.3}$$

where L_s is the free space path loss (or power loss), $\lambda = c/f$ is the wavelength of an electromagnetic wave with a frequency of f Hz and speed of $c = 0.3 \times 10^9$ m/s, which is the speed of light. The fading of the signal strength in a free space is not due to a loss of energy, as no energy is lost in propagation of a signal through a free space, but it is due to 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 the increasing distance (radius) d.

There are two types of signal fading in mobile communication systems: large-scale *slow fading* due to the movements of the mobile user over large areas and small-scale *fast fading* due to the movements of the mobile user 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 $\lambda = c/f = 0.3 \times 10^9/10^9 = 30\,\text{cm}$).

In a mobile communication 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

$$r(t) = s(t) * h_c(t) \tag{19.4}$$

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

$$r(t) = m(t) \times r_0(t) \tag{19.5}$$

Figure 19.6 illustrates the variations of large-scale slow fading and small-scale fast fading signals.

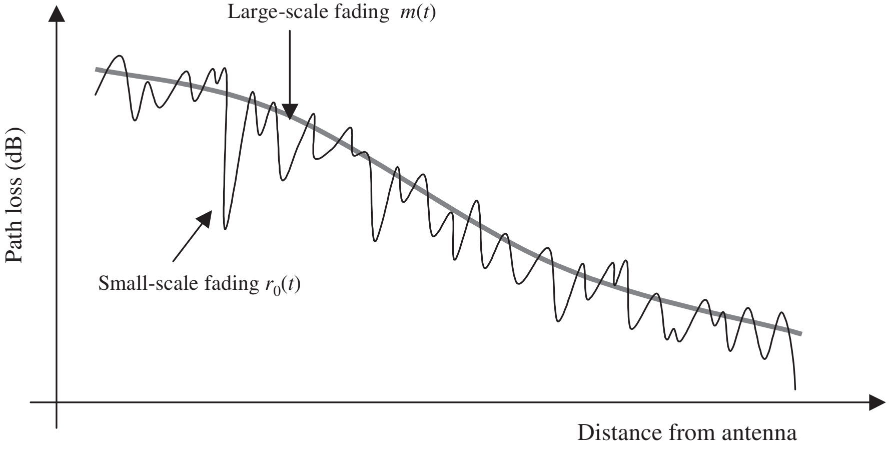


Figure 19.6 Illustration of large-scale and small-scale fading.

19.4.4 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

$$Lp(d) \propto \left(\frac{d}{d_0}\right)^n \tag{19.6}$$

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

$$10 \log L_p(d) = 10 \log L_s(d_0) + 10\, n \log\left(\frac{d}{d_0}\right) + X_\sigma \text{ dB} \tag{19.7}$$

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 a free space $n = 2$. In some urban street environments with large buildings where there may be a strong waveguide effect which contains the wave, n can be less than 2 (i.e. less propagation loss than in a free space). Generally, where there are obstructions to propagation of the energy of an electromagnetic wave, n is greater than 2.

19.4.5 Small-Scale Fast Signal Fading

Small-scale fading is due to the movements of mobile users over small distances comparable to the wavelength of the radio wave. When a radio wave signal is made of multiple reflective rays and a non-faded line-of-sight component then the fluctuations of the amplitude of the signal due to small-scale fading has 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

$$p(r) = \begin{cases} \dfrac{r}{\sigma^2} \exp\left(-\dfrac{r^2}{2\sigma^2}\right) & r \geq 0 \\ 0 & r < 0 \end{cases} \tag{19.8}$$

where r is the envelope of the amplitude of the received signal and σ^2 is its variance. Often in mobile communication 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 (ISI). 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 communication 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.

19.5 Smart Antennas – Space–Time Signal Processing

Antennas launch signals in the air in the form of electromagnetic waves. 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 does the reverse function.

The manner in which an 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. Development of more efficient and intelligent antennas is central to the efforts in improving the capacity and reliability of mobile communication systems.

Figure 19.7 shows an illustration of the radiation patterns of an omni-directional antenna and a directional antenna. Omni-directional antennas radiate electromagnetic energy equally in all directions. Although this is desirable for radio/TV broadcast applications, for personal mobile communication applications it is inefficient and wasteful of power and spectrum because at any given time the mobile user is present only at one place in space and is receiving only a very small fraction of the total radiated power. Furthermore, each user of a mobile system with an omni-directional antenna contributes to the interference with the communication of other users of mobile systems who may otherwise be accommodated on the same frequency.

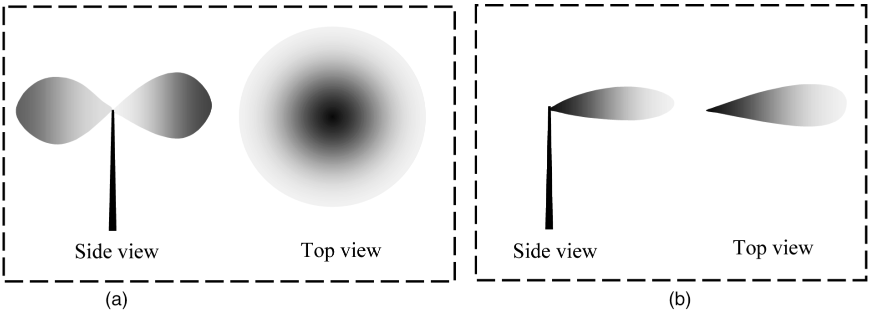


Figure 19.7 Illustration of the radiation pattern of (a) an omni-directional dipole antenna and (b) a directional antenna.

To reduce the inefficiency of omni-directional antennas, existing mobile base stations divide each cell into a number of fixed sectors with each sector covered by a fixed 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. The beam should follow the movements of the 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 transmitted signal but each with different uncorrelated noise and fading, the output SNR of the combined signals 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 17.

19.5.1 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 two main forms of smart beam-forming antennas, namely: switched antenna array and adaptive antenna array.

As illustrated in Figure 19.8, a 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 sectorised 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.

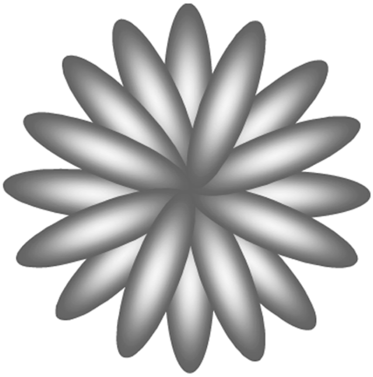


Figure 19.8 Illustration of a switched antenna where a beam is selected from a number of pre-programmed options.

Adaptive smart antenna 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.

19.5.2 Space–Time Signal Processing – Diversity Schemes

Space–time signal processing refers to the signal processing methods that utilise the signal processing possibilities offered by the transmission and or reception of several signals across time and space using multiple transmitter/receiver antennas.

The use of multiple-input multiple-output (MIMO) transmitter/receiver antennas together with adaptive signal processing schemes opens up a myriad of possibilities in the area of space–time signal processing. Signals from different sources can be arranged, combined and processed in many different ways 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 communication systems. These are:

(1) The design of the physical antennas as electromagnetic radiators. This is mainly concerned with such antenna characteristics as the radiation pattern, the main-lobe beamwidth, side-lobe levels and power efficiency.
(2) The estimation of direction of arrival of the electromagnetic pattern in beam-forming antennas using such signal processing methods as MUSIC and ESPIRIT.
(3) Development of signal processing methods, and MIMO coders/decoders, for antenna arrays in order to improve the 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, multi-path reflections, spatial directions or polarisations. 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 replicas of the messages from different routes can be processed and combined to increase the signal-to-noise ratio. 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 communication channels and increase the channel capacity.

A number of diversity schemes are described in the literature, and we will briefly consider some basic diversity schemes here.

(1) 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. A form of space diversity recently proposed is co-operative communication in which it is proposed that different handsets from different mobile users can co-operate and act as a virtual antenna arrays to receive multiple copies of a signal through a diversity of propagation routes in space.
(2) Polarisation diversity: Antennas can transmit either a horizontally polarised wave or a vertically polarised 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 polarisations.
(3) Angle diversity: Since the received signal arrives at the antenna via several paths, each with a different angle of arrival, the use of 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.
(4) 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 are uncorrelated. For frequency separations of more than several times the coherence bandwidth the signal fading would be essentially uncorrelated.
(5) Time diversity: When the same data are sent over the channel at different time instants, the received signals can be uncorrelated if the time separations are large enough. The required time separation is

at least as great as the reciprocal of the fading bandwidth, which is two times the speed of the mobile station divided by the wavelength. Hence, the time separation is inversely proportional to the speed of the mobile station. For a stationary device, time diversity is useless. This is in contrast to all of the other diversity types discussed above because they are independent of the speed of the mobile station.

19.6 Summary

The reduction of noise, interference, channel distortion and multi-path effects are some of the most important challenges in mobile communication. Cellular architecture and the use of smart antennas are the main methods for improving the capacity and spectral efficiency in mobile communication systems. This chapter provided an overview of some of the main issues in modelling and reduction of noise and interference in wireless communication systems.

As the demand for multimedia communication 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 would 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, Inc.

Lee Y.W.C. (1989) *Mobile Cellular Communications*. New York, McGraw-Hill.

Liberti J.C. and Rappaport T.S. (1999) *Smart Antennas for Wireless Communications: IS-95 and Third-Generation CDMA Applications*. Prentice Hall.

Padgett J.E., Günther C.G. and Hattori T. (1995) Overview of Wireless Personal Communications, *IEEE Communications Magazine*, 28–41.

Pahlavan K. and Levesque A.H. (1996) *Wireless Information Networks*, John Wiley & Sons, Inc.

Paulraj A., Nabar R., and Gore D. (2003) *Introduction to Space–Time Signal Processing*. Cambridge University Press.

Rappaport T.S. (1996) *Wireless Communications: Principles and Practice*, 2nd edn, Prentice-Hall.

Schilling D.L. (1994) Wireless Communications Going into the 21st Century, *IEEE Trans*. on Vehicular Technology, **43**(3): 645–651.

Shannon C.E. (1948), A Mathematical Theory of Communication, *Bell System Tech. J*., **27**, in two parts: 379–423, 623–656, July, October.

Sklar B. (1997) Rayleigh Fading Channels in Mobile Digital Communication Systems; Part I: Characterization; Part II: Mitigation', *IEEE Communication Magazine*, **35**(7): 90–109, July.

Van Trees H.L. (1898) *Detection, Estimation and Modulation Theory*, Part I. New York, John Wiley & Sons, Inc.

Index

Advanced Digital Signal Processing and Noise Reduction Fourth Edition Saeed V. Vaseghi