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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

Advanced Digital Signal Processing and Noise Reduction

Second Edition

Advanced Digital Signal Processing and Noise Reduction

Second Edition

SAEED V. VASEGHI
Professor of Communications and Signal Processing,
Department of Electronics and Computer Engineering,
Brunel University, UK

JOHN WILEY & SONS, LTD
Chichester · New York · Weinheim · Brisbane · Singapore · Toronto

First Edition published in 1996 jointly by John Wiley & Sons, Ltd. and B. G. Teubner as Advanced Signal Processing and Digital Noise Reduction.

Baffins Lane, Chichester,
West Sussex, PO19 1UD, England

National 01243 779777
International (+44) 1243 779777

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

Visit our Home Page on http://www.wiley.co.uk or http://www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

Weinheim • Brisbane • Singapore • Toronto

Library of Congress Cataloging-in-Publication Data
Vaseghi, Saeed V.
Advanced digital signal processing and noise reduction / Saeed V. Vaseghi.—2nd ed.
p.cm.
Includes bibliographical references and index.
ISBN 0-471-62692-9 (alk.paper)
1. Signal processing. 2. Electronic noise. 3. Digital Filters (Mathematics) I. Title.
TK5102.9. V37 2000
621.382'2—dc21

00-032091

British Library Cataloguing in Publication Data

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

ISBN 0 471 62692 9

Produced from PostScript files supplied by the author.
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire.
This book is printed on acid-free paper responsibly manufactured from sustainable forestry, in which at least two trees are planted for each one used for paper production.

To my parents

With thanks to Peter Rayner, Ben Milner, Charles Ho and Aimin Chen

CONTENTS

PREFACE

Signal processing theory plays an increasingly central role in the development of modern telecommunication and information processing systems, and has a wide range of applications in multimedia technology, audio-visual signal processing, cellular mobile communication, adaptive network management, radar systems, pattern analysis, medical signal processing, financial data forecasting, decision making systems, etc. 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 is an important part 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.

This book is organised in 15 chapters.

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

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

Chapter 3 provides an introduction to the theory and applications of probability models and stochastic signal processing. The chapter begins with an introduction to random signals, stochastic processes, probabilistic models and statistical measures. The concepts of stationary, 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 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 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 10 considers interpolation of a sequence of unknown samples. This chapter begins with a study of the ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples, and the factors that affect interpolation. Interpolators are divided into two

categories: polynomial and statistical interpolators. A general form of polynomial interpolation as well as its special forms (Lagrange, Newton, Hermite and cubic spline interpolators) are 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 11 considers spectral subtraction. 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 concludes with an implementation of spectral subtraction for signal restoration.

Chapters 12 and 13 cover the modelling, detection and removal of impulsive noise and transient noise pulses. In Chapter 12, impulsive noise is modelled as a binary–state 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 14 covers echo cancellation. The chapter begins with an introduction to telephone line echoes, and considers line echo suppression and adaptive line echo cancellation. Then the problem of acoustic echoes and acoustic coupling between loudspeaker and microphone systems are considered. The chapter concludes with a study of a sub-band echo cancellation system

Chapter 15 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.

Saeed Vaseghi
June 2000

FREQUENTLY USED SYMBOLS AND ABBREVIATIONS

AWGN	Additive white Gaussian noise
ARMA	Autoregressive moving average process
AR	Autoregressive process
$\boldsymbol{A}$	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
bps	Bits per second
$b(m)$	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
$f_X(\boldsymbol{x})$	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 $\boldsymbol{Y}$
$f_{X;\boldsymbol{\theta}}(\boldsymbol{x};\boldsymbol{\theta})$	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
$\boldsymbol{h}$	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

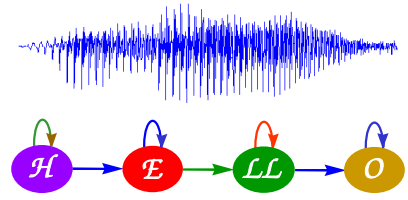
$H^{inv}(f)$	Inverse channel frequency response
$\boldsymbol{H}$	Observation matrix, Distortion matrix
$\mathbf{I}$	Identity matrix
$\boldsymbol{J}$	Fisher's information matrix
$\|\boldsymbol{J}\|$	Jacobian of a transformation
$\boldsymbol{K}(m)$	Kalman gain matrix
LSE	Least square error
LSAR	Least square AR interpolation
λ	Eigenvalue
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
MAP	Maximum a posterior estimate
MA	Moving average process
ML	Maximum likelihood estimate
MMSE	Minimum mean squared error estimate
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
$\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{\boldsymbol{xx}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}})$	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)
pdf	Probability density function
pmf	Probability mass function
$P_X(\boldsymbol{x}_i)$	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 $\boldsymbol{y}_j$
$P_{X\|Y}(\boldsymbol{x}_i \| \boldsymbol{y}_j)$	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
$\boldsymbol{r}_{xx}(m)$	Autocorrelation vector
$\boldsymbol{R}_{xx}$	Autocorrelation matrix of signal $\boldsymbol{x}(m)$
$\boldsymbol{R}_{xy}$	Cross–correlation matrix
$\boldsymbol{s}$	State sequence
$\boldsymbol{s}^{ML}$	Maximum–likelihood state sequence
SNR	Signal-to-noise ratio
SINR	Signal-to-impulsive noise ratio
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 of $x(m)$
$X(f,t)$	Time-frequency spectrum of $x(m)$
$\boldsymbol{X}$	Clean signal matrix
$\boldsymbol{X}^{\mathrm{H}}$	Hermitian transpose of $\boldsymbol{X}$
$y(m)$	Noisy signal
$\boldsymbol{y}(m)$	Noisy signal vector
$\hat{\boldsymbol{y}}(m\|m-i)$	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

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

1



INTRODUCTION

Signal processing is concerned with the modelling, detection, identification and utilisation of patterns and structures in a signal process. Applications of signal processing methods include audio hi-fi, digital TV and radio, cellular mobile phones, voice recognition, vision, radar, sonar, geophysical exploration, medical electronics, and in general any system that is concerned with the communication or processing of information. Signal processing theory plays a central role in the development of digital telecommunication and automation systems, and in efficient and optimal transmission, reception and decoding of information. Statistical signal processing theory provides the foundations for modelling the distribution of random signals and the environments in which the signals propagate. Statistical models are applied in signal processing, and in decision-making systems, for extracting information from a signal that may be noisy, distorted or incomplete. This chapter begins with a definition of signals, and a brief introduction to various signal processing methodologies. We consider several key applications of digital signal processing in adaptive noise reduction, channel equalisation, pattern classification/recognition, audio signal coding, signal detection, spatial processing for directional reception of signals, Dolby noise reduction and radar. The chapter concludes with an introduction to sampling and conversion of continuous-time signals to digital signals.

1.1 Signals and Information

A signal can be defined as the variation of a quantity by which information is conveyed regarding the state, the characteristics, the composition, the trajectory, the course of action or the intention of the signal source. *A signal is a means to convey information.* The information conveyed in a signal may be used by humans or machines for communication, forecasting, decision-making, control, exploration etc. Figure 1.1 illustrates an information source followed by a system for signalling the information, a communication channel for propagation of the signal from the transmitter to the receiver, and a signal processing unit at the receiver for extraction of the information from the signal. In general, there is a mapping operation that maps the information $I(t)$ to the signal $x(t)$ that carries the information, this mapping function may be denoted as $T[\cdot]$ and expressed as

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

For example, in human speech communication, the voice-generating mechanism provides a means for the talker to map each word into a distinct acoustic speech signal that can propagate to the listener. To communicate a word w, the talker generates an acoustic signal realisation of the word; this acoustic signal x(t) may be contaminated by ambient noise and/or distorted by a communication channel, or impaired by the speaking abnormalities of the talker, and received as the noisy and distorted signal $y(t)$. In addition to conveying the spoken word, the acoustic speech signal has the capacity to convey information on the speaking characteristic, accent and the emotional state of the talker. The listener extracts these information by processing the signal $y(t)$.

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

Signal processing methods are central to efficient communication, and to the development of intelligent man/machine interfaces in such areas as

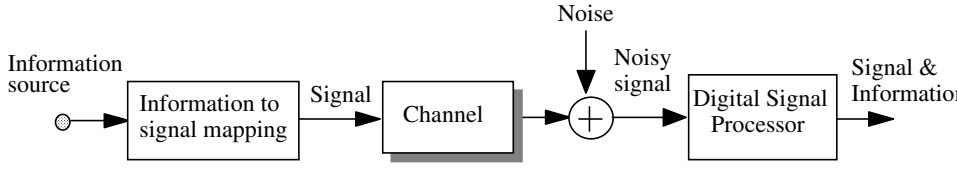


Figure 1.1 Illustration of a communication and signal processing system.

speech and visual pattern recognition for multimedia systems. In general, digital signal processing is concerned with two broad areas of information theory:

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

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

1.2 Signal Processing Methods

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

Depending on the method used, digital signal processing algorithms can be categorised into one or a combination of four broad categories. These are non–parametric signal processing, model-based signal processing, Bayesian statistical signal processing and neural networks. These methods are briefly described in the following.

1.2.1 Non–parametric Signal Processing

Non–parametric methods, as the name implies, do *not* utilise a parametric model of the signal generation or a model of the statistical distribution of the signal. The signal is processed as a waveform or a sequence of digits. Non–parametric methods are not specialised to any particular class of signals, they are broadly applicable methods that can be applied to any signal regardless of the characteristics or the source of the signal. The drawback of these methods is that they do not utilise the distinct characteristics of the signal process that may lead to substantial

improvement in performance. Some examples of non–parametric methods include digital filtering and transform-based signal processing methods such as the Fourier analysis/synthesis relations and the discrete cosine transform. Some non–parametric methods of power spectrum estimation, interpolation and signal restoration are described in Chapters 9, 10 and 11.

1.2.2 Model-Based Signal Processing

Model-based signal processing methods utilise a parametric model of the signal generation process. The parametric model normally describes the predictable structures and the expected patterns in the signal process, and can be used to forecast the future values of a signal from its past trajectory. Model-based methods normally outperform 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 Signal Processing

The fluctuations of a purely random signal, or the distribution of a class of random signals in the signal space, cannot be modelled by a predictive equation, but can be described in terms of the statistical average values, and modelled by a probability distribution function in a multidimensional signal space. For example, as described in Chapter 8, a linear prediction model driven by a random signal can model 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. Chapter 4 describes the Bayesian inference methodology and the estimation of random processes observed in noise.

1.2.4 Neural Networks

Neural networks are combinations of relatively simple 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 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 in this book.

1.3 Applications of Digital Signal Processing

In recent years, the development and commercial availability of increasingly powerful and affordable digital computers has been accompanied by the development of advanced digital signal processing algorithms for a wide variety of applications such as noise reduction, 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 1.2 shows a broad categorisation of some DSP applications. This section provides a review of several key applications of digital signal processing methods.

1.3.1 Adaptive Noise Cancellation and Noise Reduction

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

$$y(m) = x(m) + n(m) \tag{1.2}$$

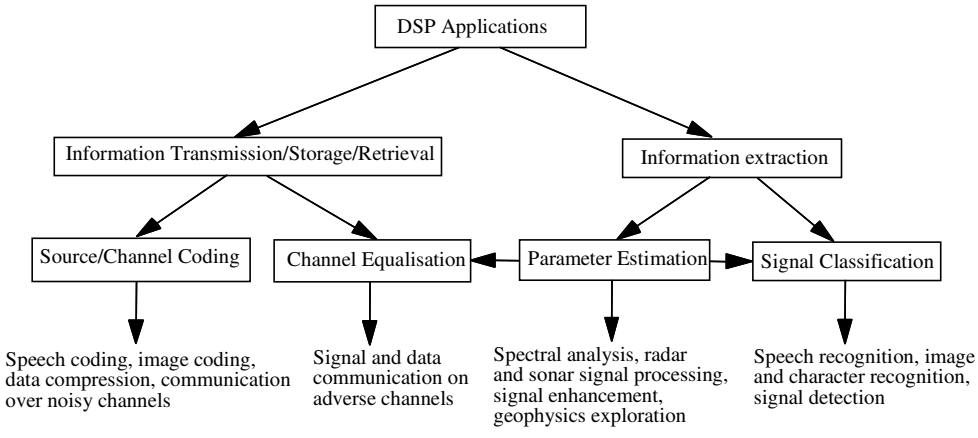


Figure 1.2 A classification of the applications of digital signal processing.

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 1.3 shows a two-input adaptive noise cancellation system for enhancement of noisy speech. In this system a directional microphone takes

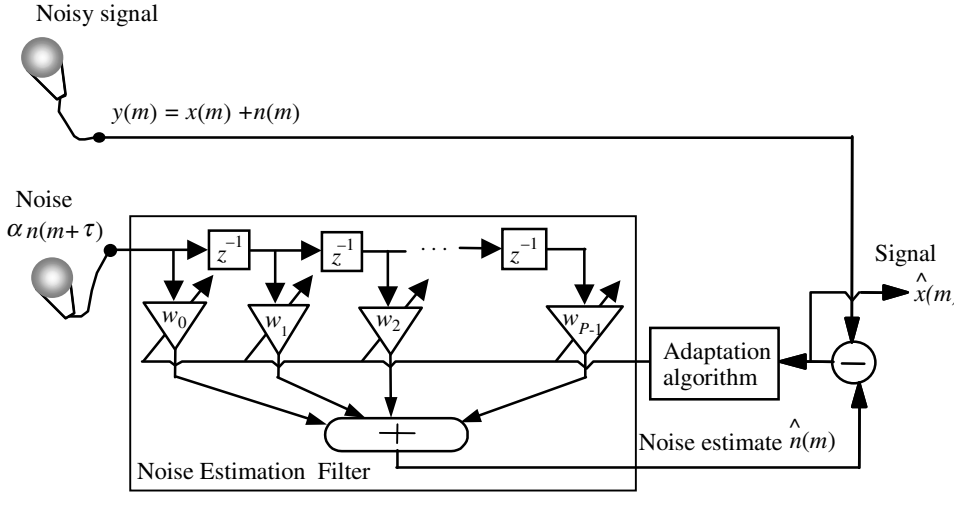


Figure 1.3 Configuration of a two-microphone adaptive noise canceller.

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.

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 1.4 shows a bank of Wiener filters for reducing additive noise when only the

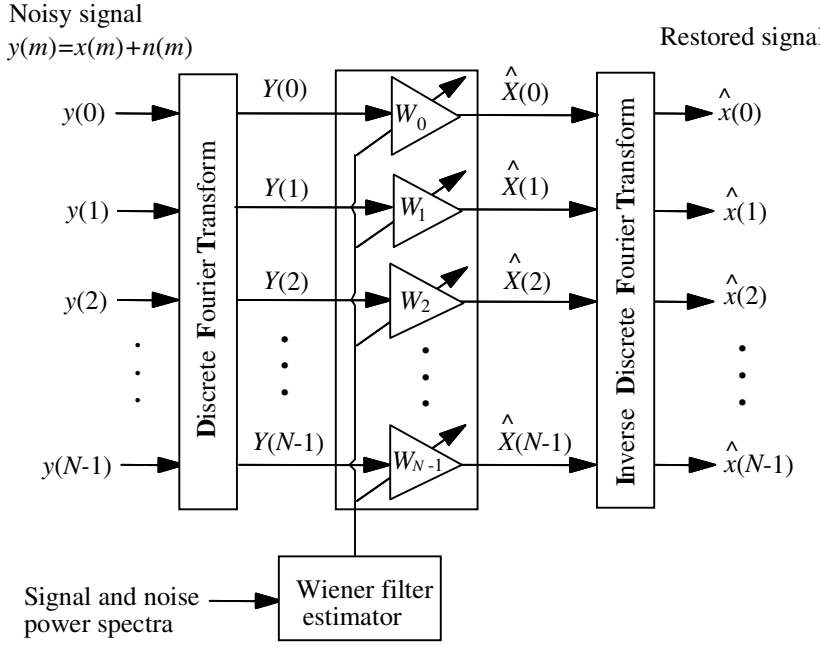


Figure 1.4 A frequency–domain Wiener filter for reducing additive noise.

noisy signal is available. The filter bank coefficients attenuate each noisy signal frequency in inverse proportion to the signal–to–noise ratio at that frequency. The Wiener filter bank coefficients, derived in Chapter 6, are calculated from estimates of the power spectra of the signal and the noise processes.

1.3.2 Blind Channel Equalisation

Channel equalisation is the recovery of a signal distorted in transmission through a communication channel with a non-flat magnitude 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 1.5 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

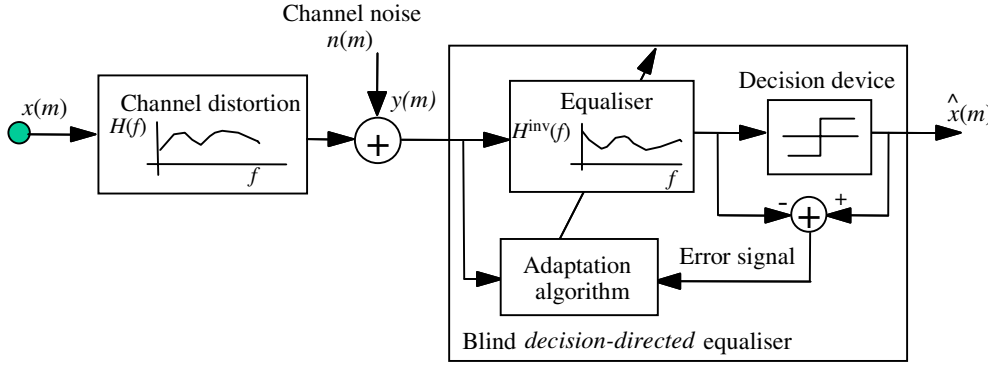


Figure 1.5 Configuration of a decision-directed blind channel equaliser.

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 15.

1.3.3 Signal Classification and Pattern Recognition

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

To design a classifier; a set of models 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 code book, of waveforms, each representing the prototype for one class of signals. A more complete model for each class of signals takes the form of a probability distribution function. In the classification phase, a signal is labelled with the nearest or the most likely class. For example, in communication of a binary bit stream over a band-pass channel, the binary phase–shift keying (BPSK) scheme signals the bit "1" using the waveform $A_c \sin \omega_c t$ and the bit "0" using $-A_c \sin \omega_c t$. At the receiver, the decoder has the task of classifying and labelling the received noisy signal as a "1" or a "0". Figure 1.6 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

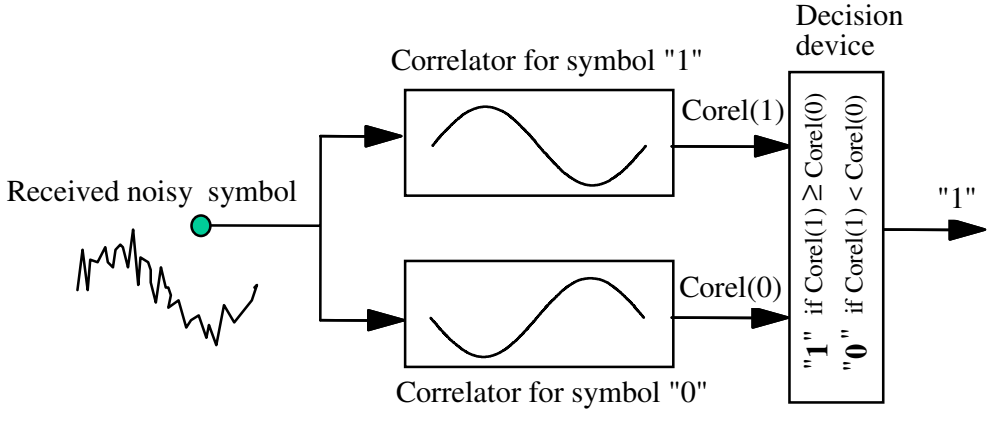


Figure 1.6 A block diagram illustration of the classifier in a binary phase-shift keying demodulation.

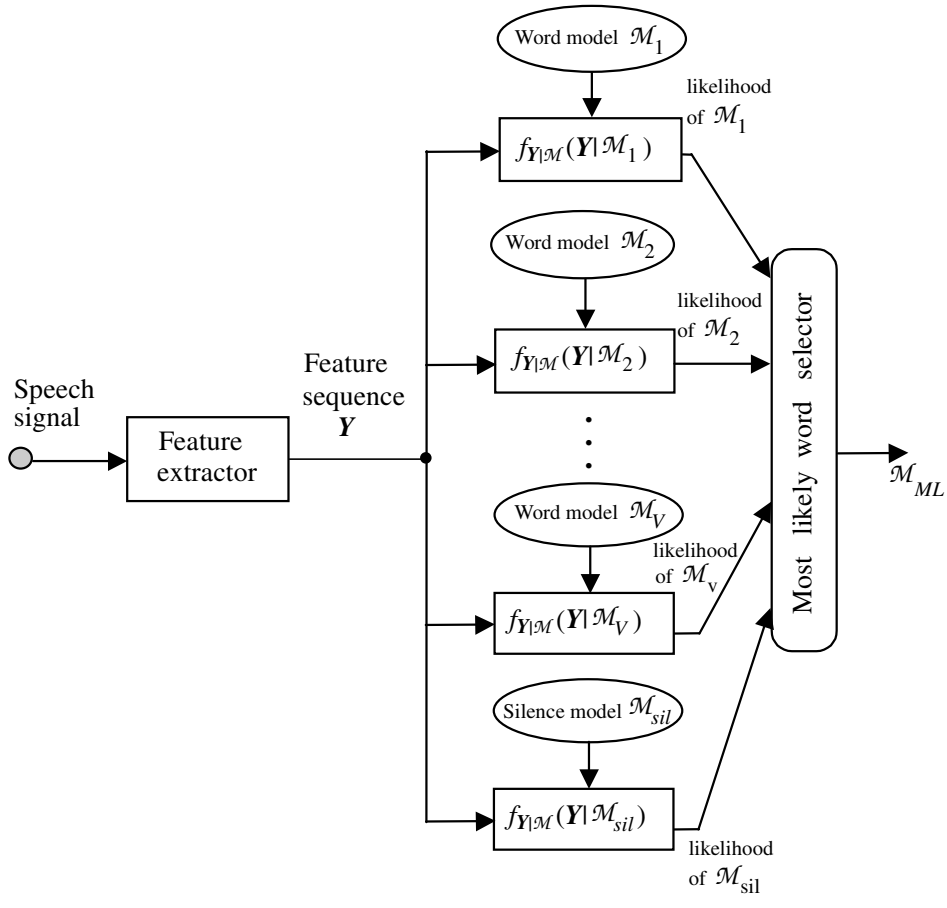


Figure 1.7 Configuration of speech recognition system, $f(\mathbf{Y}|\mathcal{M}_i)$ is the likelihood of the model $\mathcal{M}_i$ given an observation sequence $\mathbf{Y}$.

states for the bit "1" and the bit "0". The decoder correlates the unlabelled input signal with each of the two candidate symbols and selects the candidate that has a higher correlation with the input.

Figure 1.7 illustrates the use of a classifier in a limited–vocabulary, isolated-word speech recognition system. Assume there are V words in the vocabulary. For each word a model is trained, on many different examples of the spoken word, to capture the average characteristics and the statistical variations of the word. The classifier has access to a bank of V+1 models, one for each word in the vocabulary and an additional model for the silence periods. In the speech recognition phase, the task is to decode and label an

acoustic speech feature sequence, representing an unlabelled spoken word, as one of the V likely words or silence. For each candidate word the classifier calculates a probability score and selects the word with the highest score.

1.3.4 Linear Prediction Modelling of Speech

Linear predictive models are widely used in speech processing applications such as low–bit–rate speech coding in cellular telephony, speech enhancement and speech recognition. Speech is generated by inhaling air into the lungs, and then exhaling it through the vibrating glottis cords and the vocal tract. The random, noise-like, air flow from the lungs is spectrally shaped and amplified by the vibrations of the glottal cords and the resonance of the vocal tract. The effect of the vibrations of the glottal cords and the vocal tract is to introduce a measure of correlation and predictability on the random variations of the air from the lungs. Figure 1.8 illustrates a model for speech production. The source models the lung and emits a random excitation signal which is filtered, first by a pitch filter model of the glottal cords and then by a model of the vocal tract.

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

$$\hat{x}(m)=\sum_{k=1}^{P} a_k x(m-k) \tag{1.3}$$

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

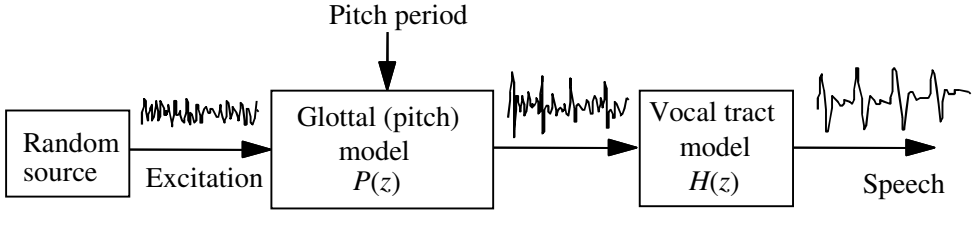


Figure 1.8 Linear predictive model of speech.

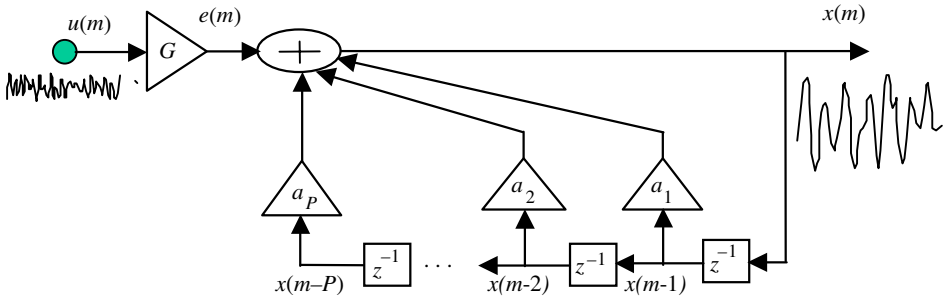


Figure 1.9 Illustration of a signal generated by an all-pole, linear prediction model.

prediction error $e(m)$, i.e. the difference between the actual sample $x(m)$ and its predicted value $\hat{x}(m)$, is defined as

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

The prediction error $e(m)$ may also be interpreted as the random excitation or the so-called innovation content of $x(m)$. From Equation (1.4) a signal generated by a linear predictor can be synthesised as

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

Equation (1.5) describes a speech synthesis model illustrated in Figure 1.9.

1.3.5 Digital Coding of Audio Signals

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

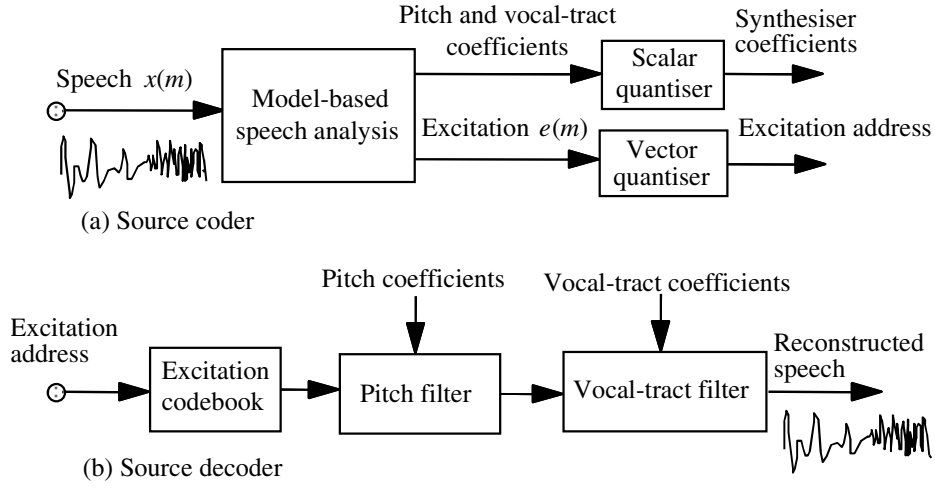


Figure 1.10 Block diagram configuration of a model-based speech coder.

applications such as cellular telephony; and transform-based coders used in high–quality coding of speech and digital hi-fi audio.

Figure 1.10 shows a simplified block diagram configuration of a speech coder–synthesiser of the type used in digital cellular telephone. The speech signal is modelled as the output of a filter excited by a random signal. The random excitation models the air exhaled through the lung, and the filter models the vibrations of the glottal cords and the vocal tract. At the transmitter, speech is segmented into blocks 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 synthesis the speech. At the receiver, the model parameters and the excitation are used to reconstruct the speech.

A transform-based coder is shown in Figure 1.11. The aim of transformation is to convert the signal into a form where it lends itself to a more convenient and useful interpretation and manipulation. In Figure 1.11 the input signal is transformed to the frequency domain using a filter bank, or a discrete Fourier transform, or a discrete cosine transform. Three main advantages of coding a signal in the frequency domain are:

(a) 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.

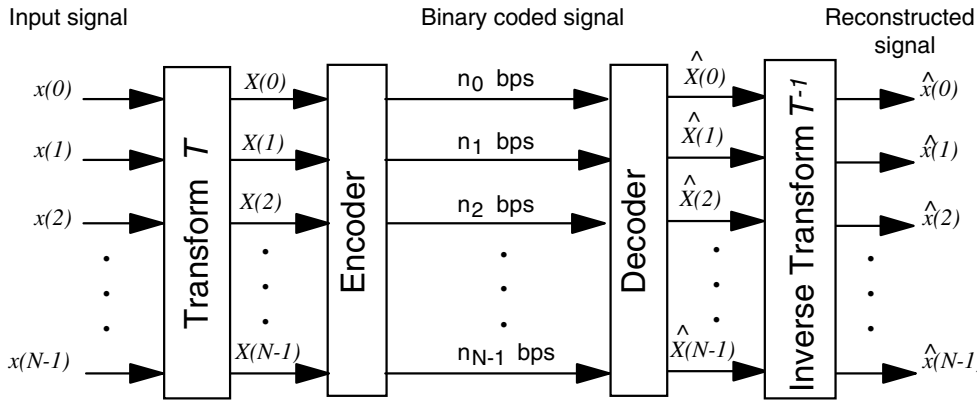


Figure 1.11 Illustration of a transform-based coder.

(b) 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.
(c) 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.

1.3.6 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

$$y(m) = b(m)x(m) + n(m) \tag{1.6}$$

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

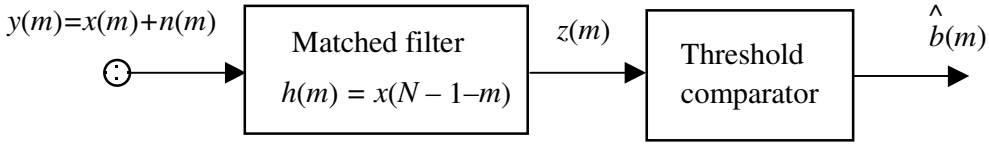


Figure 1.12 Configuration of a matched filter followed by a threshold comparator for detection of signals in noise.

can be used to detect the signal as shown in Figure 1.12. The impulse response $h(m)$ of the matched filter for detection of a signal $x(m)$ is the time-reversed version of $x(m)$ given by

$$h(m)=x(N-1-m) \qquad 0 \le m \le N-1 \tag{1.7}$$

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

$$z(m)=\sum_{m=0}^{N-1} h(m-k)\,y(m) \tag{1.8}$$

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

$$\hat{b}(m)=\begin{cases}1 & \text{if } z(m) \ge \text{threshold} \\ 0 & \text{otherwise}\end{cases} \tag{1.9}$$

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. Table1.1 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

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

Table 1.1 Four possible outcomes in a signal detection problem.

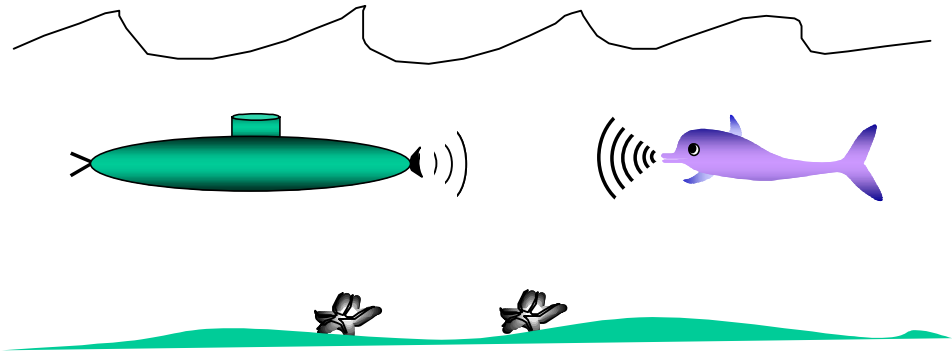


Figure 1.13 Sonar: detection of objects using the intensity and time delay of reflected sound waves.

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

$$\mathcal{R}(\text{Threshold} = \theta) = P_{\text{False Alarm}}(\theta) + P_{\text{Miss}}(\theta) \tag{1.10}$$

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

1.3.7 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 incident at a particular spatial angle are passed through, whereas those arriving from other directions are attenuated. Beam-forming is used in radar and sonar signal processing (Figure 1.13) to steer the reception of signals towards a desired direction, and in speech processing for reducing the effects of ambient noise.

To explain the process of beam-forming consider a uniform linear array of sensors as illustrated in Figure 1.14. The term *linear array* implies that the array of sensors is spatially arranged in a straight line and with equal spacing d between the sensors. Consider a sinusoidal far–field plane wave with a frequency F_0 propagating towards the sensors at an incidence angle of θ as illustrated in Figure 1.14. The array of sensors samples the incoming

wave as it propagates in space. The time delay for the wave to travel a distance of d between two adjacent sensors is given by

$$\tau = \frac{d \sin\theta}{c} \tag{1.11}$$

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

$$\varphi = 2\pi \frac{\tau}{T_0} = 2\pi F_0 \frac{d \sin\theta}{c} \tag{1.12}$$

where T_0 is the period of the sine wave. By inserting appropriate corrective

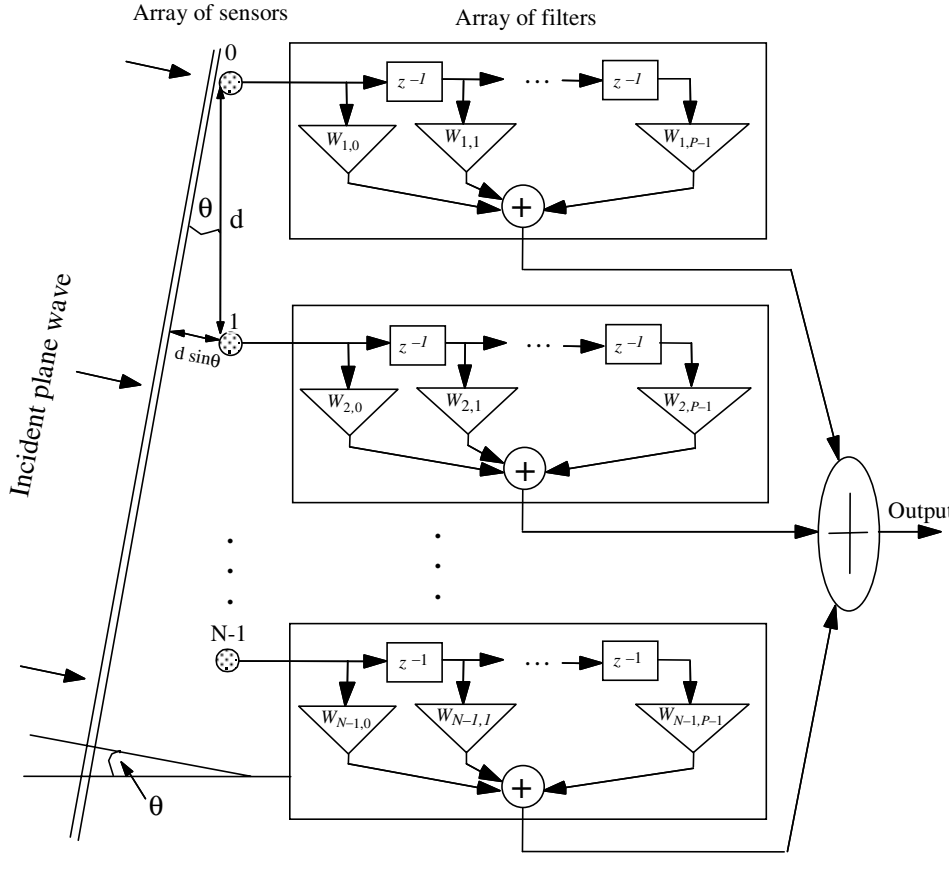


Figure 1.14 Illustration of a beam-former, for directional reception of signals.

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 1.14 illustrates a beam-former as an array of digital filters arranged in space. The filter array acts as a two–dimensional space–time signal processing system. The space filtering allows the beam-former to be steered 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.

1.3.8 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 part 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 out each other in order to avoid processing distortion.

Dolby has 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

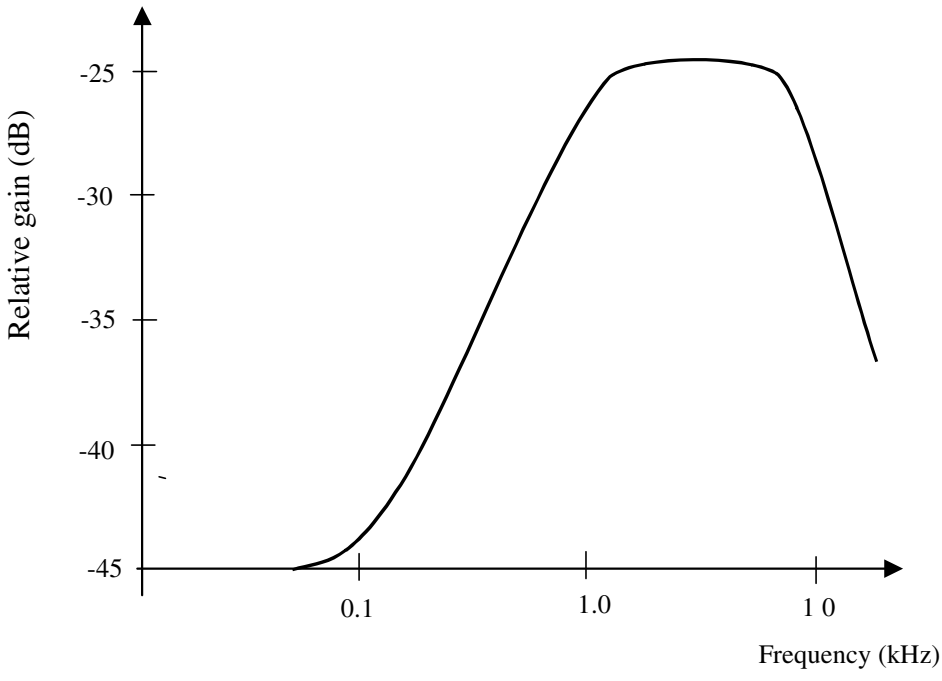


Figure 1.15 Illustration of the pre-emphasis response of Dolby-C: upto 20 dB boost is provided when the signal falls 45 dB below maximum recording level.

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 Figure1.15.

1.3.9 Radar Signal Processing: Doppler Frequency Shift

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

$$x(t) = A(t)\cos\{\omega_0 [t - 2r(t)/c]\} \tag{1.13}$$

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

$$r(t) = r_0 + \dot{r}t + \frac{1}{2!}\ddot{r}t^2 + \frac{1}{3!}\dddot{r}t^3 + \cdots \tag{1.14}$$

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

$$r(t) \approx r_0 + \dot{r}t \tag{1.15}$$

Substituting Equation (1.15) in Equation (1.13) yields

$$x(t) = A(t)\cos[(\omega_0 - 2\dot{r}\omega_0/c)t - 2\omega_0 r_0/c] \tag{1.16}$$

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

$$\omega_d = 2\dot{r}\omega_0/c \tag{1.17}$$

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

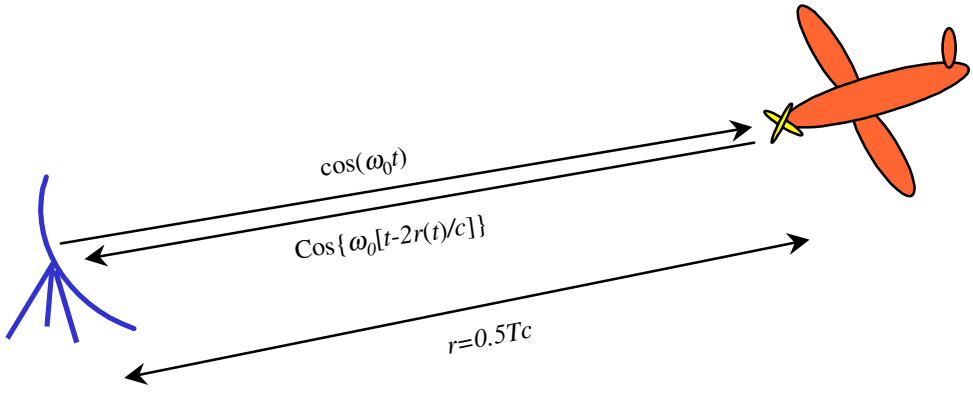


Figure 1.16 Illustration of a radar system.

object is moving away from the radar then the distance $r(t)$ is increasing, $\dot{r}$ is positive, and a decrease in the frequency is observed. Thus the frequency analysis of the reflected signal can reveal information on the direction and speed of the object. The distance r_0 is given by

$$r_0 = 0.5T \times c \tag{1.18}$$

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

1.4 Sampling and Analog–to–Digital Conversion

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

$$\delta(m) = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \tag{1.19}$$

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

$$x(m) = \sum_{k=-\infty}^{\infty} x(k)\delta(m-k) \tag{1.20}$$

Figure 1.17 illustrates a discrete-time signal. Many random processes, such as speech, music, radar and sonar generate signals that are continuous in

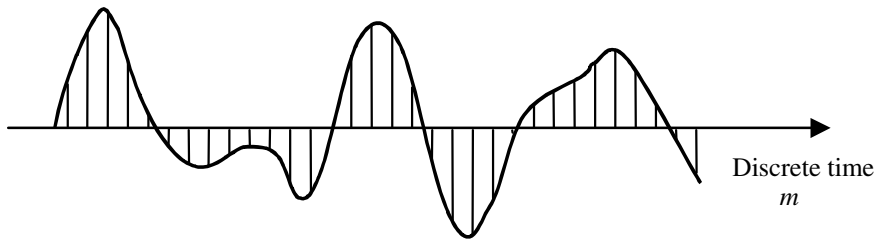


Figure 1.17 A discrete-time signal and its envelope of variation with time.

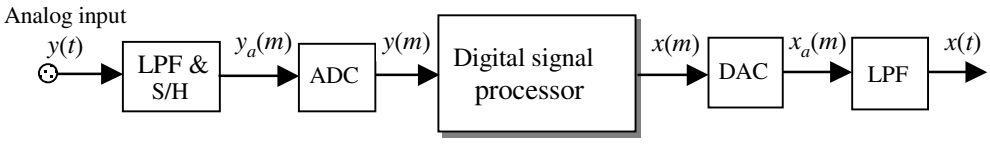


Figure 1.18 Configuration of a digital signal processing system.

time and continuous in amplitude. Continuous signals are termed analog because their fluctuations with time are analogous to the variations of the signal source. For digital processing, analog signals are sampled, and each sample is converted into an n-bit digit. The digitisation process should be performed such that the original signal can be recovered from its digital version with no loss of information, and with as high a fidelity as is required in an application. Figure 1.18 illustrates a block diagram configuration of a digital signal processor with an analog input. The low-pass filter removes out–of–band signal frequencies above a pre-selected range. The sample–and–hold (S/H) unit periodically samples the signal to convert the continuous-time signal into a discrete-time signal.

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

1.4.1 Time-Domain Sampling and Reconstruction of Analog Signals

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

$$x_{\text{sampled}}(t) = x(t)\,p(t) = \sum_{m=-\infty}^{\infty} x(t)\delta(t - mT_s) \tag{1.21}$$

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

$$p(t) = \sum_{m=-\infty}^{\infty} \delta(t - mT_s) \tag{1.22}$$

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

$$P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \tag{1.23}$$

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

$$X_{\text{sampled}}(f) = FT[x(t).p(t)] = X(f) * P(f) = \sum_{k=-\infty}^{\infty} \delta(f - kF_s) \tag{1.24}$$

where the operator $FT[.]$ denotes the Fourier transform. In Equation (1.24) the convolution of a signal spectrum $X(f)$ with each impulse $\delta(f - kF_s)$, shifts $X(f)$ and centres it on kF_s. *Hence, as expressed in Equation (1.24), the sampling of a signal x(t) results in a periodic repetition of its spectrum* $X(f)$ *centred on frequencies* $0, \pm F_s, \pm 2F_s, \ldots$. When the sampling frequency is higher than twice the maximum frequency content of the signal, then the repetitions of the signal spectra are separated as shown in Figure 1.19. In this case, the analog signal can be recovered by passing the sampled signal through an analog low-pass filter with a cut-off frequency of F_s. If the sampling frequency is less than $2F_s$, then the adjacent repetitions of the spectrum overlap and the original spectrum cannot be recovered. The distortion, due to an insufficiently high sampling rate, is irrevocable and is known as *aliasing*. This observation is the basis of the *Nyquist sampling theorem* which states: a band-limited continuous-time signal, with a highest

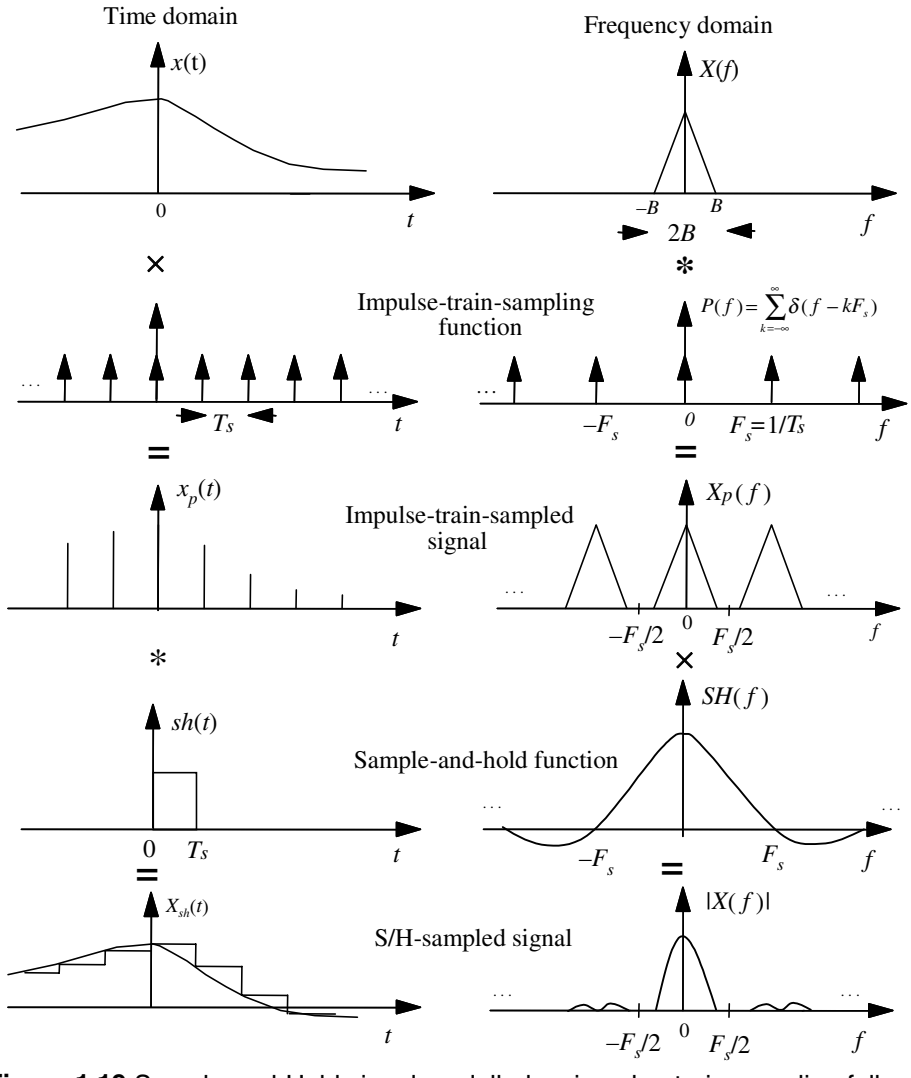


Figure 1.19 Sample-and-Hold signal modelled as impulse-train sampling followed by convolution with a rectangular pulse.

frequency content (bandwidth) of B Hz, can be recovered from its samples provided that the sampling speed $F_s > 2B$ samples per second.

In practice sampling is achieved using an electronic switch that allows a capacitor to charge up or down to the level of the input voltage once every T_s seconds as illustrated in Figure 1.20. 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 Figure1.19.

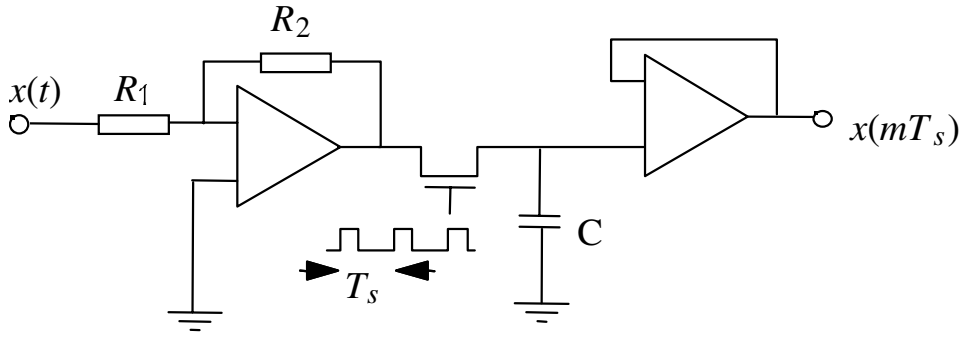


Figure 1.20 A simplified sample-and-hold circuit diagram.

1.4.2 Quantisation

For digital signal processing, continuous-amplitude samples from the sample-and-hold are quantised and mapped into n-bit binary digits. For quantisation to n bits, the amplitude range of the signal is divided into 2^n discrete levels, and each sample is quantised to the nearest quantisation level, and then mapped to the binary code assigned to that level. Figure 1.21 illustrates the quantisation of a signal into 4 discrete levels. Quantisation is a many-to-one mapping, in that all the values that fall within the continuum of a quantisation band are mapped to the centre of the band. The mapping between an analog sample $x_a(m)$ and its quantised value $x(m)$ can be expressed as

$$x(m) = Q[x_a(m)] \tag{1.25}$$

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

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

$$e(m) = x(m) - x_a(m) \tag{1.26}$$

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

$$\mathcal{E}\left[e^2(m)\right]=\int_{-\Delta/2}^{\Delta/2} f_E(e(m))\,e^2(m)\,de(m)=\frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2(m)\,de(m)$$
$$=\frac{\Delta^2}{12}=\frac{V^2 2^{-2n}}{3} \tag{1.27}$$

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

$$SQNR(n)=10\log_{10}\left(\frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]}\right)=10\log_{10}\left(\frac{P_{\text{Signal}}}{V^2 2^{-2n}/3}\right)$$
$$=10\log_{10}3-10\log_{10}\left(\frac{V^2}{P_{\text{Signal}}}\right)+10\log_{10}2^{2n} \tag{1.28}$$
$$=4.77-\alpha+6n$$

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

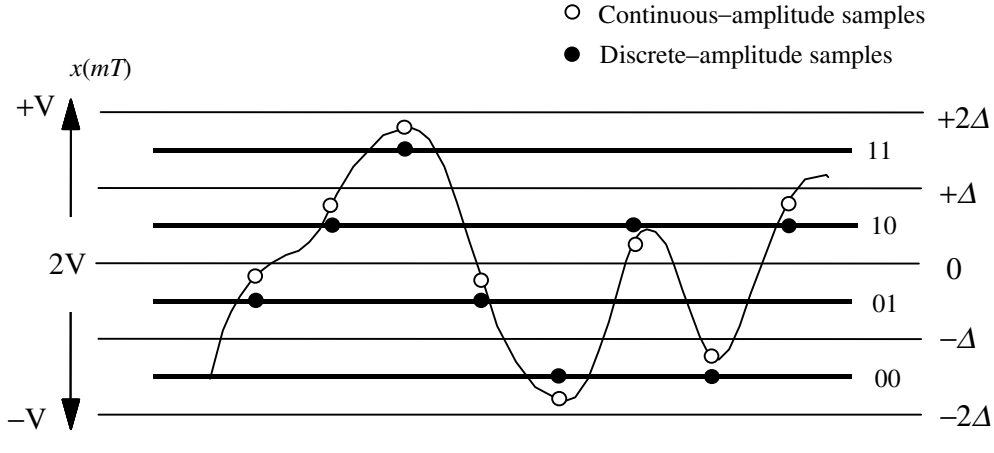


Figure 1.21 Offset-binary scalar quantisation

Bibliography

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

DAVENPORT W.B. and ROOT W.L. (1958) An Introduction to the Theory of Random Signals and Noise. McGraw-Hill, New York.

EPHRAIM Y. (1992) Statistical Model Based Speech Enhancement Systems. Proc. IEEE, **80**, **10**, pp. 1526–1555.

GAUSS K.G. (1963) Theory of Motion of Heavenly Bodies. Dover, New York.

GALLAGER R.G. (1968) Information Theory and Reliable Communication. Wiley, New York.

HAYKIN S. (1991) Adaptive Filter Theory. Prentice-Hall, Englewood Cliffs, NJ.

HAYKIN S. (1985) Array Signal Processing. 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**, pp. 35–45.

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

LIM J.S. (1983) Speech Enhancement. Prentice Hall, Englewood Cliffs, NJ.

LUCKY R.W., SALZ J. and WELDON E.J. (1968) Principles of Data Communications. McGraw-Hill, New York.

KUNG S.Y. (1993) Digital Neural Networks. Prentice-Hall, Englewood Cliffs, NJ.

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

OPPENHEIM A.V. and SCHAFER R.W. (1989) Discrete-Time Signal Processing. Prentice-Hall, Englewood Cliffs, NJ.

PROAKIS J.G., RADER C.M., LING F. and NIKIAS C.L. (1992) Advanced Signal Processing. Macmillan, New York.

RABINER L.R. and GOLD B. (1975) Theory and Applications of Digital Processing. Prentice-Hall, Englewood Cliffs, NJ.

RABINER L.R. and 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.

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

VAN-TREES H.L. (1971) Detection, Estimation and Modulation Theory. Parts I, II and III. Wiley New York.

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

WILSKY A.S. (1979) Digital Signal Processing, Control and Estimation Theory: Points of Tangency, Areas of Intersection and Parallel Directions. MIT Press, Cambridge, MA.

WIDROW B. (1975) Adaptive Noise Cancelling: Principles and Applications. Proc. IEEE, **63**, pp. 1692-1716.

WIENER N. (1948) Extrapolation, Interpolation and Smoothing of Stationary Time Series. MIT Press, Cambridge, MA.

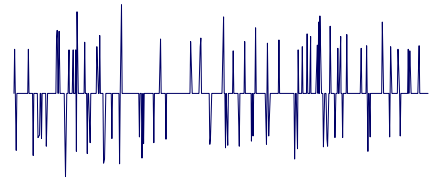
WIENER N. (1949) Cybernetics. MIT Press, Cambridge, MA.

ZADEH L.A. and DESOER C.A. (1963) Linear System Theory: The State-Space Approach. McGraw-Hill, NewYork.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

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 a signal that conveys information regarding the source of the noise. For example, the noise from a car engine conveys information regarding the state of the engine. The sources of noise are many, and vary from audio frequency acoustic noise emanating from moving, vibrating or colliding sources such as revolving machines, moving vehicles, computer fans, keyboard clicks, wind, rain, etc. to radio-frequency electromagnetic noise that can interfere with the transmission and reception of voice, image and data over the radio-frequency spectrum. Signal distortion is the term often used to describe a systematic undesirable change in a signal and refers to changes in a signal due to the non–ideal characteristics of the transmission channel, reverberations, echo and missing samples.

Noise and distortion are the main limiting factors in communication and measurement systems. Therefore the modelling and removal of the effects of noise and distortion 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 signals cannot be isolated from noise and distortion. In this chapter, we study the characteristics and modelling of several different forms of noise.

2.1 Introduction

Noise may be defined as any unwanted signal that interferes with the communication, measurement 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 variety of noise that could degrade the quality of communication, such as acoustic background noise, thermal noise, electromagnetic radio-frequency noise, co-channel interference, radio-channel distortion, echo and processing noise. Noise can cause transmission errors and may even disrupt a communication process; hence noise processing is an important 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. Depending on its source, a noise can be classified into a number of categories, indicating the broad physical nature of the noise, as follows:

(a) 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 cars, air-conditioners, computer fans, traffic, people talking in the background, wind, rain, etc.
(b) Electromagnetic noise: present at all frequencies and in particular at the radio frequencies. All electric devices, such as radio and television transmitters and receivers, generate electromagnetic noise.
(c) 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.
(d) Channel distortions, echo, and fading: due to non-ideal characteristics of communication channels. Radio channels, such as those at microwave frequencies used by cellular mobile phone operators, are particularly sensitive to the propagation characteristics of the channel environment.
(e) Processing noise: the noise that results from the digital/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.

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

(a) Narrowband noise: a noise process with a narrow bandwidth such as a 50/60 Hz 'hum' from the electricity supply.
(b) White noise: purely random noise that has a flat power spectrum. White noise theoretically contains all frequencies in equal intensity.
(c) Band-limited white noise: a noise with a flat spectrum and a limited bandwidth that usually covers the limited spectrum of the device or the signal of interest.
(d) Coloured noise: non-white noise or any wideband noise whose spectrum has a non-flat shape; examples are pink noise, brown noise and autoregressive noise.
(e) Impulsive noise: consists of short-duration pulses of random amplitude and random duration.
(f) Transient noise pulses: consists of relatively long duration noise pulses.

2.2 White Noise

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

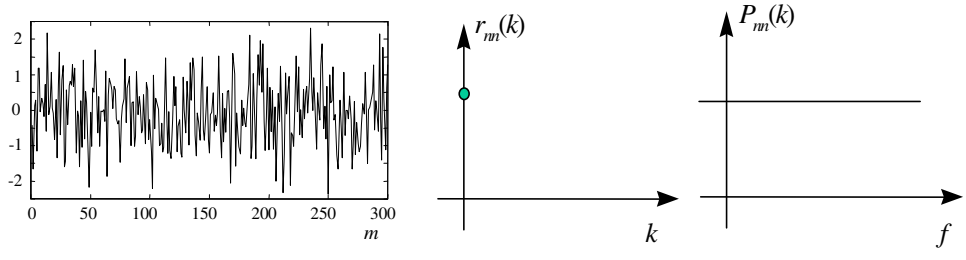


Figure 2.1 Illustration of (a) white noise, (b) its autocorrelation, and (c) its power spectrum.

The autocorrelation function of a continuous-time zero-mean white noise process with a variance of σ^2 is a delta function given by

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

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

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

Equation (2.2) shows that a white noise has a constant power spectrum.

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 is given by

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

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

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

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

2.3 Coloured Noise

Although the concept of white noise provides a reasonably realistic and mathematically convenient and useful approximation to some predominant noise processes encountered in 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, has a non-white predominantly low-frequency spectrum. Also, a white noise passing through a channel is "coloured" by the shape of the channel spectrum. Two classic varieties of coloured noise are so-called pink noise and brown noise, shown in Figures 2.2 and 2.3.

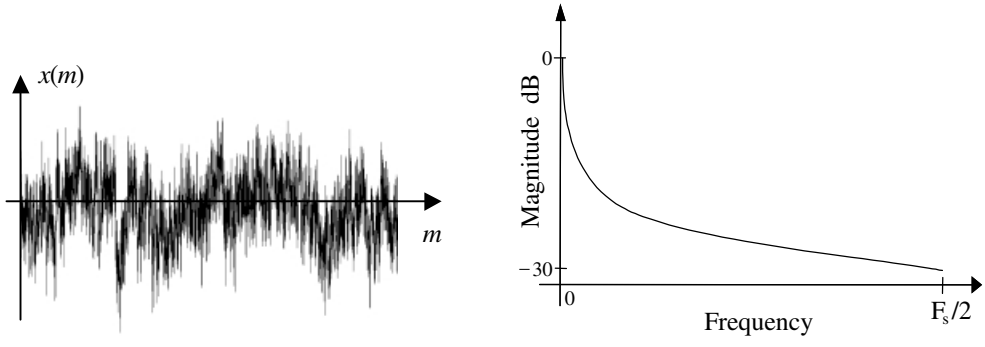


Figure 2.2 (a) A pink noise signal and (b) its magnitude spectrum.

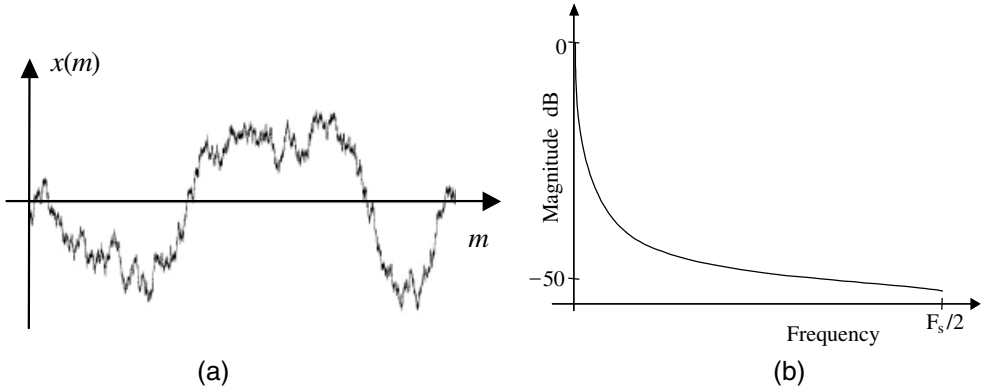


Figure 2.3 (a) A brown noise signal and (b) its magnitude spectrum.

2.4 Impulsive Noise

Impulsive noise consists of short-duration "on/off" noise pulses, caused by a variety of sources, such as switching noise, adverse channel environment in a communication system, drop-outs or surface degradation of audio recordings, clicks from computer keyboards, etc. Figure 2.4(a) shows an ideal impulse and its frequency spectrum. In communication systems, a real impulsive-type noise has a duration that is normally more than one sample long. For example, in the context of audio signals, short-duration, sharp pulses, of up to 3 milliseconds (60 samples at a 20 kHz sampling rate) may be considered as impulsive noise. Figures 2.4(b) and (c) illustrate two examples of short-duration pulses and their respective spectra.

In a communication system, an impulsive noise originates at some point in time and space, and then propagates through the channel to the receiver. The received noise is 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, in response to a large-amplitude impulse, exhibit a non-linear characteristic.

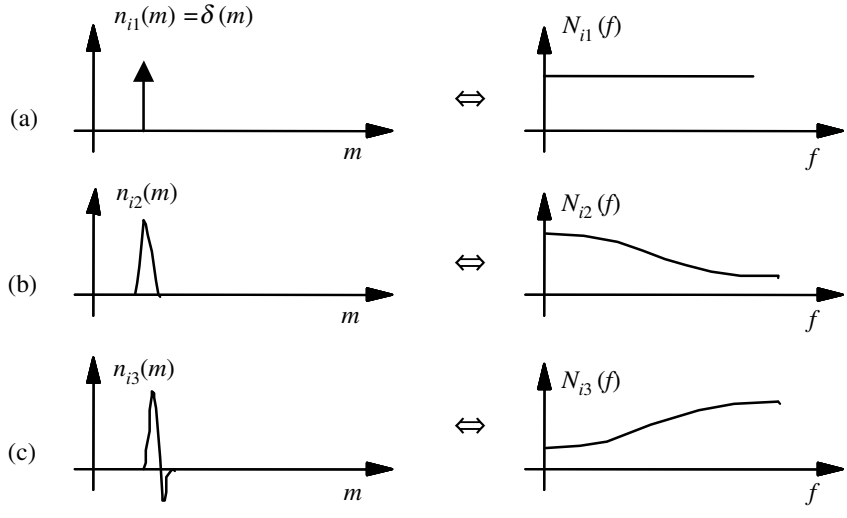


Figure 2.4 Time and frequency sketches of: (a) an ideal impulse, (b) and (c) short-duration pulses.

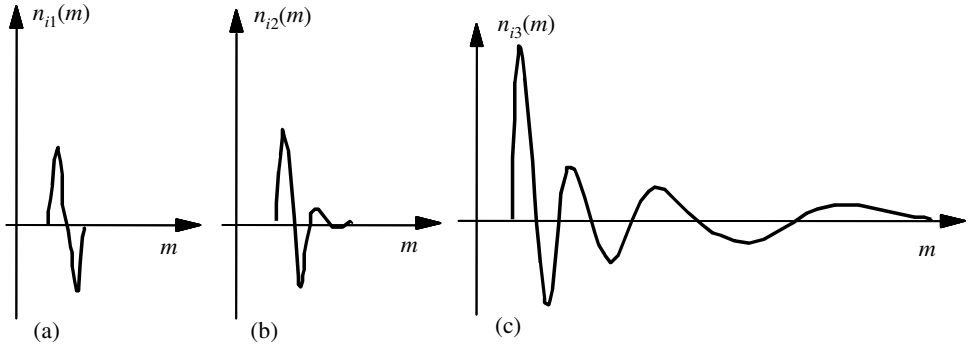


Figure 2.5 Illustration of variations of the impulse response of a non-linear system with the increasing amplitude of the impulse.

Figure 2.5 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 2.5(c) a large impulse excitation has generated a decaying transient pulse. These variations may be attributed to the non-linear characteristics of the playback mechanism.

2.5 Transient Noise Pulses

Transient noise pulses often consist of a relatively short sharp initial pulse followed by decaying low-frequency oscillations as shown in Figure 2.6. The initial pulse is usually due to some external or internal impulsive interference, whereas the oscillations are often due to the resonance of the

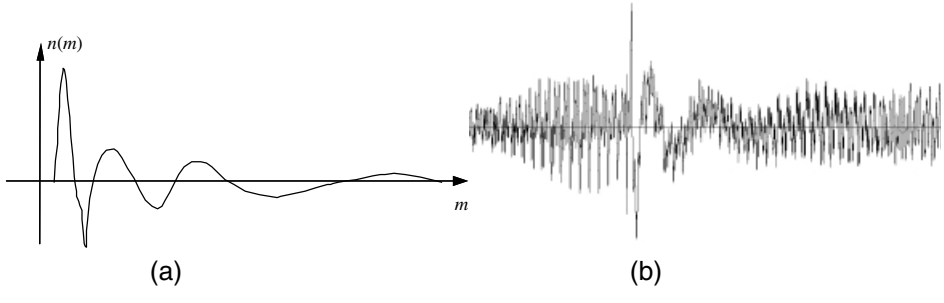


Figure 2.6 (a) A scratch pulse and music from a gramophone record. (b) The averaged profile of a gramophone record scratch pulse.

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 should be able to characterize the transient noise pulses with a similar degree of consistency as in characterizing the channels through which the pulses propagate.

As an illustration of the shape of a transient noise pulse, consider the scratch pulses from a damaged gramophone record shown in Figures 2.6(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 2.6(b), a typical scratch pulse waveform often exhibits two distinct regions:

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

Note in Figure 2.6(b) that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the 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 detain in Chapter 13.

2.6 Thermal Noise

Thermal noise, also referred to as Johnson noise (after its discoverer J. B. Johnson), is generated by the random movements of thermally energised particles. 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, 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=1.38×10^{-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

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

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

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

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

The term shot noise arose from the analysis of random variations in the emission of electrons from the cathode of a vacuum tube. Discrete electron particles in a current flow arrive at random times, and therefore there will be fluctuations about the average particle flow. The fluctuations in the rate of particle flow constitutes the shot noise. Other instances of shot noise are 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. The concept of randomness of the rate of emission or arrival of particles implies that shot noise can be modelled by a Poisson distribution. When the average number of arrivals during the observing time is large, the fluctuations will approach a Gaussian distribution. Note that whereas thermal noise is due to "unforced" random movement of particles, shot noise happens in a forced directional flow of particles.

Now consider an electric current as the flow of discrete electric charges. If the charges act independently of each other the fluctuating current is given by

$$I_{\text{Noise}}(\text{rms}) = (2eI_{\text{dc}}B)^{1/2} \tag{2.9}$$

where $e = 1.6\times10^{-19}$ coulomb is the electron charge, and B is the measurement bandwidth. For example, a "steady" current I_{dc} of 1 amp in a bandwidth 1 MHz has an rms fluctuation of 0.57 microamps. 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 Electromagnetic Noise

Virtually every electrical device that generates, consumes or transmits power is a potential source of electromagnetic noise and 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 electrical storms.

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

Electrostatic fields are generated by the presence of voltage, with or without current flow. Fluorescent lighting is one of the more common sources of electrostatic noise. Magnetic fields are created either by the flow of electric current or by the presence of permanent magnetism. Motors and transformers are examples of the former, and the Earth's magnetic field is an instance of the latter. In order for noise voltage to be developed in a conductor, magnetic lines of flux must be cut by the conductor. Electric 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.

2.9 Channel Distortions

On propagating through a channel, signals are shaped and distorted by the frequency response and the attenuating 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

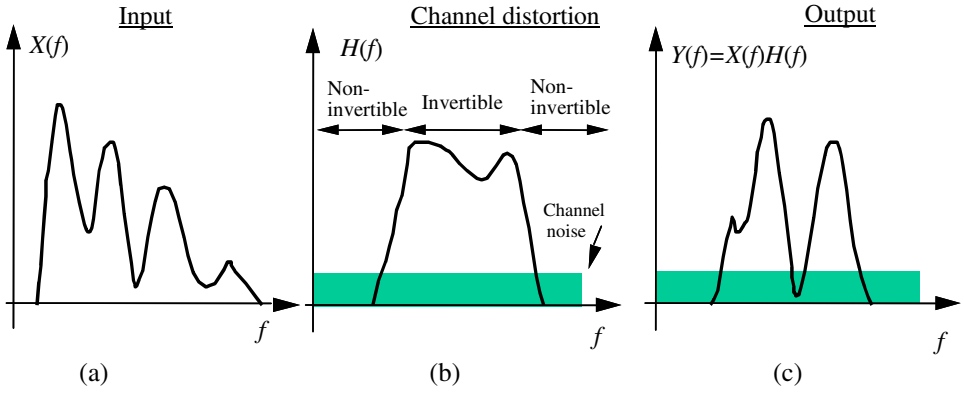


Figure 2.7 Illustration of channel distortion: (a) the input signal spectrum, (b) the channel frequency response, (c) the channel output.

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 equalization are essential components of modern digital communication systems. Channel equalization 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 2.7 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 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.

2.10 Modelling Noise

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

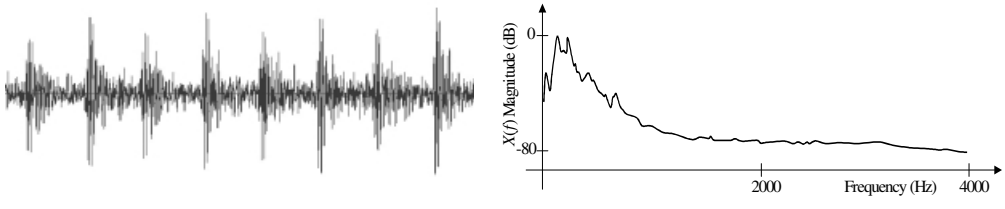
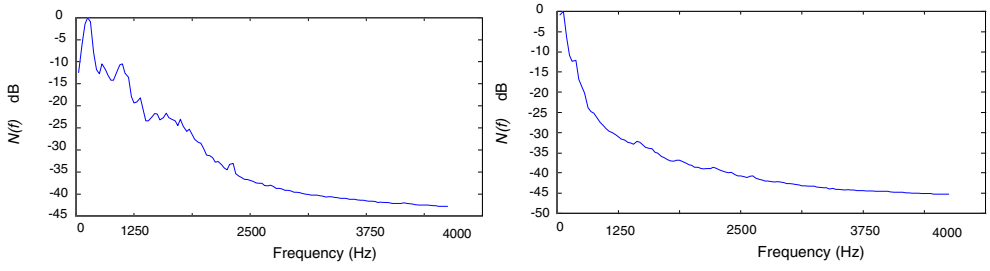


Figure 2.8 Illustration of: (a) the time-waveform of a drill noise, and (b) the frequency spectrum of the drill noise.



(a) (b)

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

analysis. Figure 2.8 illustrates the noise from an electric drill, which, as expected, has a periodic structure. The spectrum of the drilling noise shown in Figure 2.8(a) reveals that most of the noise energy is concentrated in the lower-frequency part of the spectrum. In fact, it is true of most audio signals and noise that they have a predominantly low-frequency spectrum. However, it must be noted that the relatively lower-energy high-frequency part of audio signals plays an important part in conveying sensation and quality. Figures 2.9(a) and (b) show examples of the spectra of car noise recorded from a BMW and a Volvo respectively. The noise in a car is nonstationary, and varied, and may include the following sources:

(a) quasi-periodic noise from the car engine and the revolving mechanical parts of the car;
(b) noise from the surface contact of wheels and the road surface;
(c) noise from the air flow into the car through the air ducts, windows, sunroof, etc;
(d) noise from passing/overtaking vehicles.

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

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

2.10.1 Additive White Gaussian Noise Model (AWGN)

In 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.

2.10.2 Hidden Markov Model for Noise

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

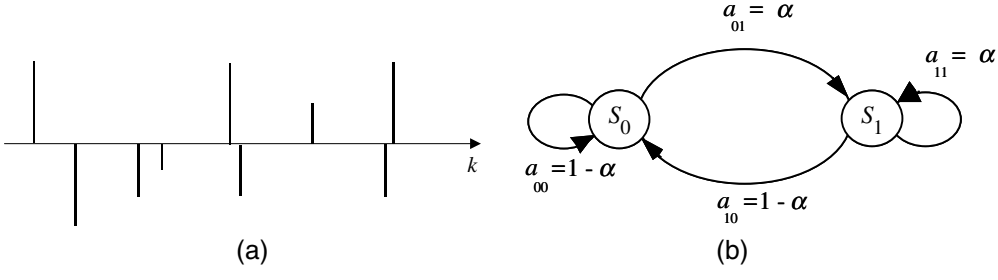


Figure 2.10 (a) An impulsive noise sequence. (b) A binary-state model of impulsive noise.

the non-stationary character of the noise.

An example of a non-stationary noise is the impulsive noise of Figure 2.10(a). Figure 2.10(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 11 for modelling impulsive noise, and in Chapter 14 for channel equalisation.

Bibliography

BELL D.A. (1960) Electrical Noise and Physical Mechanism. Van Nostrand, London.

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

DAVENPORT W.B. and ROOT W.L. (1958) An Introduction to the Theory of Random Signals and Noise. McGraw-Hill, New York.

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

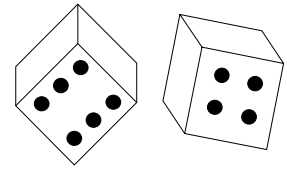
SCHWARTZ M. (1990) Information Transmission, Modulation and Noise. 4th Ed., McGraw-Hill, New York.

EPHRAIM Y. (1992) Statistical Model Based Speech Enhancement Systems. Proc. IEEE **80**, **10**, pp. 1526–1555.

VAN-TREES H.L. (1971) Detection, Estimation and Modulation Theory. Parts I, II and III. Wiley, New York.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)



3

The small probability of collision of the Earth and a comet can become very great in adding over a long sequence of centuries. It is easy to picture the effects of this impact on the Earth. The axis and the motion of rotation have changed, the seas abandoning their old position...

Pierre-Simon Laplace

PROBABILITY MODELS

Probability models form the foundation of information theory. Information itself is quantified in terms of the logarithm of probability. Probability models are used to characterise and predict the occurrence of random events in such diverse areas of applications as predicting the number of telephone calls on a trunk line in a specified period of the day, road traffic modelling, weather forecasting, financial data modelling, predicting the effect of drugs given data from medical trials, etc. In signal processing, probability models are used to describe the variations of random signals in applications such as pattern recognition, signal coding and signal estimation. This chapter begins with a study of the basic concepts of random signals and stochastic processes and the models that are used for the characterisation of random processes. Stochastic processes are classes of signals whose fluctuations in time are partially or completely random, such as speech, music, image, time-varying channels, noise and video. Stochastic signals are completely described in terms of a probability model, but can also be characterised with relatively simple statistics, such as the mean, the correlation and the power spectrum. We study the concept of ergodic stationary processes in which time averages obtained from a single realisation of a process can be used instead of ensemble averages. We consider some useful and widely used classes of random signals, and study the effect of filtering or transformation of a signal on its probability distribution.

3.1 Random Signals and Stochastic Processes

Signals, in terms of one of their most fundamental characteristics, can be classified into two broad categories: *deterministic* signals and *random* signals. Random functions of time are often referred to as *stochastic* signals. In each class, a signal may be continuous or discrete in time, and may have continuous-valued or discrete-valued amplitudes.

A deterministic signal can be defined as one that traverses a predetermined trajectory in time and space. The exact fluctuations of a deterministic signal can be completely described in terms of a function of time, and the exact value of the signal 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 $x(t)=A\sin(2\pi ft+\phi)$.

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 from its past history. Most signals such as speech and noise are at least in part random. The concept of randomness is closely associated with the concepts of information and noise. Indeed, much of the work on the processing of random signals is concerned with the extraction of information from noisy observations. If a signal is 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 both information and noise. Although a random signal is not completely predictable, it often exhibits a set of well-defined statistical characteristic values such as the maximum, the minimum, the mean, the median, the variance and the power spectrum. A random process is described in terms of its statistics, and most completely in terms of a probability model from which all its statistics can be calculated.

Example 3.1 Figure 3.1(a) shows a block diagram model of a deterministic discrete-time signal. The model generates an output signal $x(m)$ from the P past samples as

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

where the function h_1 may be a linear or a non-linear model. A functional description of the model h_1 and the P initial sample values are all that is required to predict the future values of the signal $x(m)$. For example for a sinusoidal signal generator (or oscillator) Equation (3.1) becomes

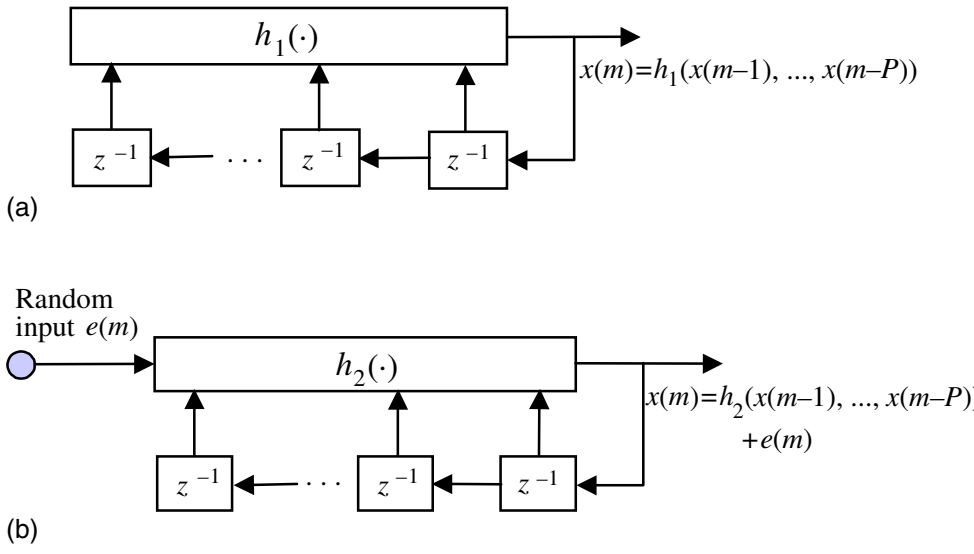


Figure 3.1 Illustration of deterministic and stochastic signal models: (a) a deterministic signal model, (b) a stochastic signal model.

$$x(m)= a\,x(m-1)- x(m-2) \tag{3.2}$$

where the choice of the parameter a=2cos(2πF_0 /F_s) determines the oscillation frequency F_0 of the sinusoid, at a sampling frequency of F_s. Figure 3.1(b) is a model for a stochastic random process given by

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

where the random input $e(m)$ models the unpredictable part of the signal $x(m)$, and the function h_2 models the part of the signal that is correlated with 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 parameters a_1 and a_2 will determine the centre frequency and the bandwidth of the process.

3.1.1 Stochastic Processes

The term "stochastic process" is broadly used to describe a random process that generates sequential signals such as speech or noise. In signal processing terminology, a stochastic process is a probability model of a class of random signals, e.g. Gaussian process, Markov process, Poisson process, etc. The classic example of a stochastic process is the so-called Brownian motion of particles in a fluid. Particles in the space of a fluid move randomly due to bombardment by fluid molecules. The random motion of each particle is a single realisation of a stochastic process. The motion of all particles in the fluid forms the collection or the space of different realisations of the process.

In this chapter, we are mainly concerned with discrete-time random processes that may occur naturally or may be obtained by sampling a continuous-time band-limited random process. The term "discrete-time stochastic process" refers to a class of discrete-time random signals, $X(m)$, characterised by a probabilistic model. Each realisation of a discrete 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.

3.1.2 The Space or Ensemble of a Random Process

The collection of all realisations of a random process is known as the ensemble, or the space, of the process. For an illustration, consider a random noise process over a telecommunication network as shown in Figure 3.2. 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 form the ensemble (or 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 the line s. The "true" statistics of a random process are obtained from the averages taken over the ensemble of many different realisations of the process. However, in many practical cases, only one realisation of a process is available. In Section 3.4, we consider the so-called ergodic processes in which time-averaged statistics, from a single realisation of a process, may be used instead of the ensemble-averaged statistics.

Notation The following notation is used in this chapter: $X(m)$ denotes a random process, the signal $x(m,s)$ is a particular realisation of the process $X(m)$, the random signal $x(m)$ is any realisation of $X(m)$, and the collection

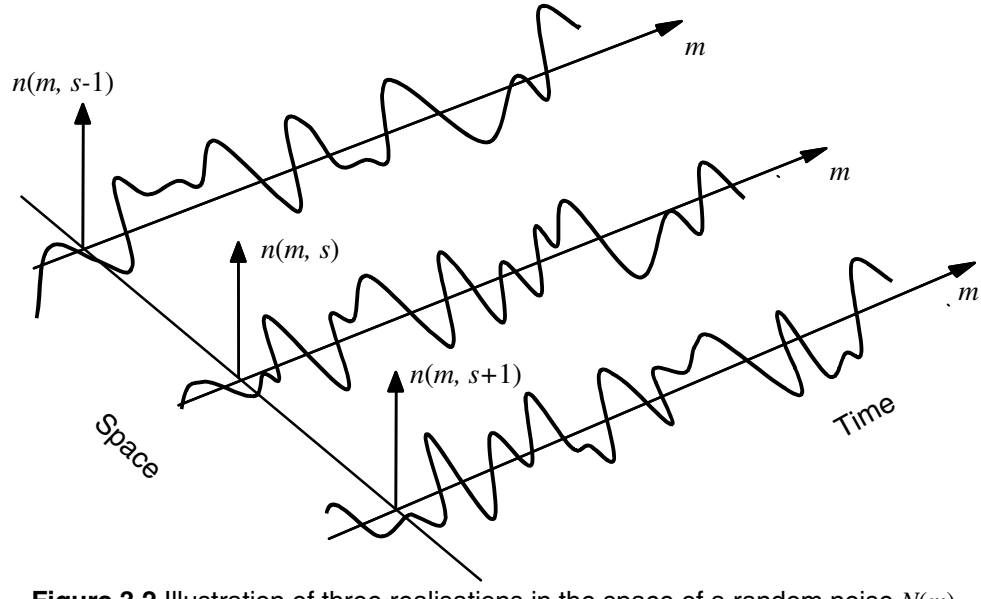


Figure 3.2 Illustration of three realisations in the space of a random noise $N(m)$.

of all realisations of $X(m)$, denoted by $\{x(m,s)\}$, form the ensemble or the space of the random process $X(m)$.

3.2 Probabilistic Models

Probability models provide the most complete mathematical description of a random process. For a fixed time instant m, the collection of sample realisations of a random process $\{x(m,s)\}$ is a random variable that takes on various values across the space s of the process. The main difference between a random variable and a random process is that the latter generates a time series. Therefore, the probability models used for random variables may also be applied to random processes. We start this section with the definitions of the probability functions for a random variable.

The space of a random variable is the collection of all the values, or outcomes, that the variable can assume. The space of a random variable can be partitioned, according to some criteria, into a number of subspaces. A subspace is a collection of signal values with a common attribute, such as a cluster of closely spaced samples, or the collection of samples with their amplitude 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

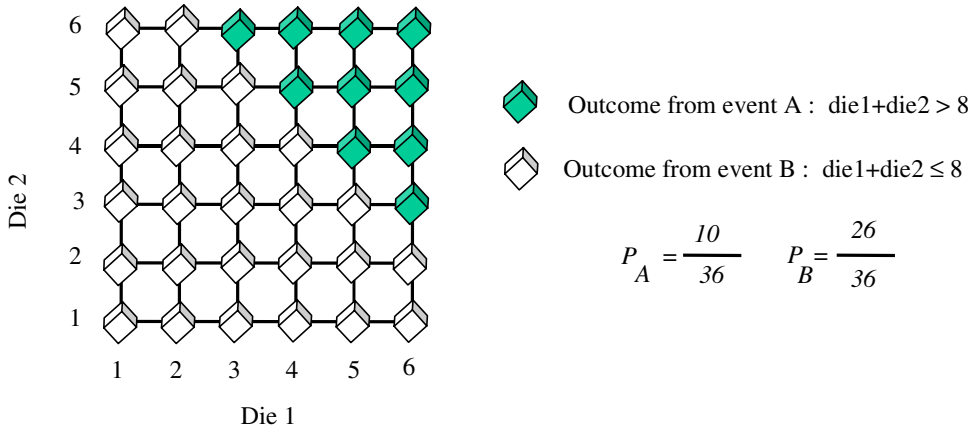


Figure 3.3 A two-dimensional representation of the outcomes of two dice, and the subspaces associated with the events corresponding to the sum of the dice being greater than 8 or, less than or equal to 8.

observed outcomes from the space of A, N_A, divided by the total number of observations:

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

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

Example 3.2 The space of two discrete numbers obtained as outcomes of throwing a pair of dice is shown in Figure 3.3. This space can be partitioned in different ways; for example, the two subspaces shown in Figure 3.3 are associated with the pair of numbers that add up to less than or equal to 8, and to greater than 8. In this example, assuming 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.

3.2.1 Probability Mass Function (pmf)

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

$$\begin{aligned} P_{X,Y}(x_i, y_j) &= P_{Y|X}(y_j | x_i) P_X(x_i) \\ &= P_{X|Y}(x_i | y_j) P_Y(y_j) \end{aligned} \tag{3.6}$$

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

$$\begin{aligned} P_X(x_i) &= \sum_{j=1}^{M} P_{X,Y}(x_i, y_j) \\ &= \sum_{j=1}^{M} P_{X|Y}(x_i | y_j) P_Y(y_j) \end{aligned} \tag{3.7}$$

where M is the number of values, or outcomes, in the space of the discrete random variable Y. From Equations (3.6) and (3.7), we have *Bayes' rule* for the conditional probability mass function, given by

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

3.2.2 Probability Density Function (pdf)

Now consider a continuous-valued random variable. A continuous-valued variable can assume an infinite number of values, and hence, the probability that it takes on a given value 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:

$$F_X(x) = Prob(X \leq x) \tag{3.9}$$

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

$$\begin{aligned}\frac{1}{\Delta} Prob(x-\Delta/2 \leq X \leq x+\Delta/2) &= \frac{1}{\Delta}[Prob(X \leq x+\Delta/2) - Prob(X \leq x-\Delta/2)] \\ &= \frac{1}{\Delta}[F_X(x+\Delta/2) - F_X(x-\Delta/2)]\end{aligned} \tag{3.10}$$

As Δ tends to zero we obtain the *probability density function* (*pdf*) as

$$\begin{aligned}f_X(x) &= \lim_{\Delta \to 0} \frac{1}{\Delta}[F_X(x+\Delta/2) - F_X(x-\Delta/2)] \\ &= \frac{\partial F_X(x)}{\partial x}\end{aligned} \tag{3.11}$$

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

$$\int_{-\infty}^{\infty} f_X(x)dx = 1 \tag{3.12}$$

The conditional and marginal probability functions and the Bayes rule, of Equations (3.6)–(3.8), also apply to probability density functions of continuous-valued variables.

Now, the probability models for random variables can also be applied to random processes. For a continuous-valued random process $X(m)$, the simplest probabilistic model is the univariate pdf $f_{X(m)}(x)$, which is the probability density function that a sample from the random process $X(m)$ takes on a value of x. A bivariate pdf $f_{X(m)X(m+n)}(x_1, x_2)$ describes the probability that the samples of the process at time instants m and $m+n$ take on the values x_1, and x_2 respectively. In general, an M-variate pdf

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

$$\int_{-\infty}^{\infty} f_{X(m_1)\cdots X(m_M)}(x_1, \ldots, x_M)\, dx_M = f_{X(m_1)\cdots X(m_{M-1})}(x_1, \ldots, x_{M-1}) \tag{3.13}$$

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

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{X(m_1)\cdots X(m_M)}(x_1, \ldots, x_M)\, dx_1 \ldots dx_M = 1 \tag{3.14}$$

The probability of a realisation 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 in the form of a conditional probability density function as

$$f_{X(m)|X(n)}(x_m | x_n) = \frac{f_{X(n)|X(m)}(x_n | x_m) f_{X(m)}(x_m)}{f_{X(n)}(x_n)} \tag{3.15}$$

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

$$f_{[X(m_1)\cdots X(m_M)|X(n_1)\cdots X(n_N)]}(x_{m_1}, \ldots, x_{m_M} | x_{n_1}, \ldots, x_{n_N}) = \prod_{i=1}^{M} f_{X(m_i)}(x_{m_i}) \tag{3.16}$$

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

$$P_{[x(m_1)\cdots x(m_M)]}(x(m_1) = x_i, \ldots, x(m_M) = x_k) \tag{3.17}$$

The probability that a discrete random process $X(m)$ takes on a value of x_m at 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

$$P_{X(m)|X(n)}(x_m|x_n)=\frac{P_{X(n)|X(m)}(x_n|x_m)P_{X(m)}(x_m)}{P_{X(n)}(x_n)} \tag{3.18}$$

and for a statistically independent process we have

$$P_{[X(m_1)\cdots X(m_M)|X(n_1)\cdots X(n_N)]}(x_{m_1},\ldots,x_{m_M}|x_{n_1},\ldots,x_{n_N})=\prod_{i=1}^{M}P_{X(m_i)}(X(m_i)=x_{m_i}) \tag{3.19}$$

3.3 Stationary and Non-Stationary Random Processes

Although the amplitude of a signal $x(m)$ fluctuates with time m, the characteristics of the process that generates the signal may be time-invariant (stationary) or time-varying (non-stationary). An example of a non-stationary process is speech, whose loudness and spectral composition changes continuously as the speaker generates various sounds. A process is stationary if the parameters of the probability model of the process are time-invariant; otherwise it is non-stationary (Figure 3.4). The stationarity property implies that all the parameters, such as the mean, the variance, the power spectral composition and the higher-order moments of the process, are time-invariant. In practice, there are various degrees of stationarity: it may be that one set of the statistics of a process is stationary, whereas another set is time-varying. For example, a random process may have a time-invariant mean, but a time-varying power.

Figure 3.4 Examples of a quasistationary and a non-stationary speech segment.

Example 3.3 In this example, we consider the *time-averaged* values of the mean and the power of: (a) a stationary signal $A\sin\omega t$ and (b) a transient signal $Ae^{-\alpha t}$.
The mean and power of the sinusoid are

$$Mean(A\sin\omega t) = \frac{1}{T}\int_T A\sin\omega t\, dt = 0, \qquad \text{constant} \tag{3.20}$$

$$Power(A\sin\omega t) = \frac{1}{T}\int_T A^2\sin^2\omega t\, dt = \frac{A^2}{2}, \qquad \text{constant} \tag{3.21}$$

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

$$Mean(Ae^{-\alpha t}) = \frac{1}{T}\int_t^{t+T} Ae^{-\alpha\tau}\, d\tau = \frac{A}{\alpha T}(1-e^{-\alpha T})e^{-\alpha t}, \qquad \text{time-varying} \tag{3.22}$$

$$Power(Ae^{-\alpha t}) = \frac{1}{T}\int_t^{t+T} A^2 e^{-2\alpha\tau}\, d\tau = \frac{A^2}{2\alpha T}(1-e^{-2\alpha T})e^{-2\alpha t}, \quad \text{time-varying} \tag{3.23}$$

In Equations (3.22) and (3.23), the signal mean and power are exponentially decaying functions of the time variable t.

Example 3.4 Consider a non-stationary signal $y(m)$ generated by a binary-state random process described by the following equation:

$$y(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \tag{3.24}$$

where $s(m)$ is a binary-valued state indicator variable and $\bar{s}(m)$ denotes the binary complement of $s(m)$. From Equation (3.24), we have

$$y(m) = \begin{cases} x_0(m) & \text{if } s(m) = 0 \\ x_1(m) & \text{if } s(m) = 1 \end{cases} \tag{3.25}$$

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

$$\begin{aligned}\mathcal{E}[y(m)|s(m)] &= \bar{s}(m)\mathcal{E}[x_0(m)] + s(m)\mathcal{E}[x_1(m)] \\ &= \bar{s}(m)\mu_{x_0} + s(m)\mu_{x_1}\end{aligned} \tag{3.26}$$

In Equation (3.26), 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

$$\begin{aligned}\mathcal{E}\left[y^2(m)|s(m)\right] &= \bar{s}(m)\mathcal{E}\left[x_0^2(m)\right] + s(m)\mathcal{E}\left[x_1^2(m)\right] \\ &= \bar{s}(m)P_{x_0} + s(m)P_{x_1}\end{aligned} \tag{3.27}$$

Although many signals are non-stationary, the concept of a stationary process has played an important role in the development of signal processing methods. Furthermore, even non-stationary signals such as speech can often 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.

3.3.1 Strict-Sense Stationary Processes

A random process $X(m)$ is stationary in a strict sense if all its distributions and statistical parameters are time-invariant. Strict-sense stationarity implies that the n^{th} order distribution is translation-invariant for all $n=1, 2,3, \ldots$:

$$\begin{aligned}&Prob[x(m_1)\leq x_1, x(m_2)\leq x_2, \ldots, x(m_n)\leq x_n)] \\ &\quad = Prob[x(m_1+\tau)\leq x_1, x(m_2+\tau)\leq x_2, \ldots, x(m_n+\tau)\leq x_n)]\end{aligned} \tag{3.28}$$

From Equation (3.28) the statistics of a strict-sense stationary process including the mean, the correlation and the power spectrum, are time-invariant; therefore we have

$$\mathcal{E}[x(m)] = \mu_x \tag{3.29}$$

$$\mathcal{E}[x(m)x(m+k)] = r_{xx}(k) \tag{3.30}$$

and

$$\mathcal{E}[|X(f,m)|^2] = \mathcal{E}[|X(f)|^2] = P_{XX}(f) \tag{3.31}$$

where μ_x, $r_{xx}(m)$ 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 $x(m)$.

3.3.2 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 so-called wide-sense stationarity. A process is said to be wide-sense stationary if the mean and the autocorrelation functions of the process are time invariant:

$$\mathcal{E}[x(m)] = \mu_x \tag{3.32}$$

$$\mathcal{E}[x(m)x(m+k)] = r_{xx}(k) \tag{3.33}$$

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.

3.3.3 Non-Stationary Processes

A random process is non-stationary if its distributions or statistics vary with time. Most stochastic processes such as video signals, audio signals, financial data, meteorological data, biomedical signals, etc., are non-stationary, because they are generated by systems whose environments and parameters vary 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 changes over time, and sometimes the change can be quite abrupt. Time-varying processes may be modelled by a combination of stationary random models as illustrated in Figure 3.5. In Figure 3.5(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 3.5(b) a time-varying process is modelled by a chain of time-invariant states, with each state having a different set of statistics or

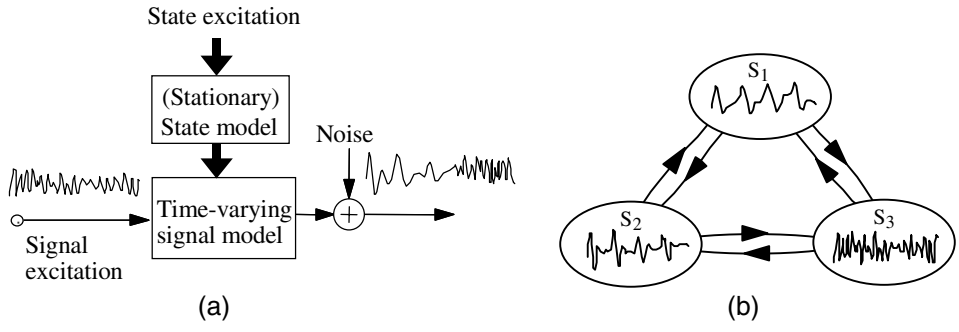


Figure 3.5 Two models for non-stationary processes: (a) a stationary process drives the parameters of a continuously time-varying model; (b) a finite-state model with each state having a different set of statistics.

probability distributions. Finite state statistical models for time-varying processes are discussed in detail in Chapter 5.

3.4 Expected Values of a Random Process

Expected values of a process play a central role in the modelling and processing of signals. 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 of the mean and the covariance of the process, and a Poisson pdf is defined in terms of the mean of the process. In signal processing applications, we often have a suitable statistical model of the process, e.g. a Gaussian pdf, and to complete the model we need the values of the expected parameters. Furthermore in many signal processing algorithms, such as spectral subtraction for noise reduction described in Chapter 11, or linear prediction described in Chapter 8, what we essentially need is an estimate of the mean or the correlation function of the process. The expected value of a function, $h(X(m_1), X(m_2), ..., X(m_M))$, of a random process X is defined as

$$\mathcal{E}[h(X(m_1),\ldots,X(m_M))] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h(x_1,\ldots,x_M) f_{X(m_1)\cdots X(m_M)}(x_1,\ldots,x_M)\, dx_1 \ldots dx_M \tag{3.34}$$

The most important, and widely used, expected values are the mean value, the correlation, the covariance, and the power spectrum.

3.4.1 The Mean Value

The mean value of a process plays an important part in signal processing and parameter estimation from noisy observations. For example, in Chapter 3 it is shown that the optimal linear estimate of a signal from a noisy observation, is an interpolation between the mean value and the observed value of the noisy signal. The mean value of a random vector $[X(m_1), ..., X(m_M)]$ is its average value across the ensemble of the process defined as

$$\mathcal{E}[X(m_1),\ldots,X(m_M)] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_1,\ldots,x_M) f_{X(m_1)\cdots X(m_M)}(x_1,\ldots,x_M)\, dx_1 \cdots dx_M \tag{3.35}$$

3.4.2 Autocorrelation

The correlation function and its Fourier transform, the power spectral density, are 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 predictive coders, equalisers, digital decoders, delay estimators, classifiers and signal restoration systems. The autocorrelation function of a random process $X(m)$, denoted by $r_{xx}(m_1,m_2)$, is defined as

$$\begin{aligned} r_{xx}(m_1,m_2) &= \mathcal{E}[x(m_1)x(m_2)] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(m_1)x(m_2) f_{X(m_1),X(m_1)}\big(x(m_1),x(m_2)\big)\, dx(m_1)\, dx(m_2) \end{aligned} \tag{3.36}$$

The autocorrelation function $r_{xx}(m_1,m_2)$ is a measure of the similarity, or the mutual relation, of the outcomes of the process X at time instants m_1 and m_2. If the outcome of a random process at time m_1 bears no relation to that at time m_2 then $X(m_1)$ and $X(m_2)$ are said to be independent or uncorrelated and $r_{xx}(m_1,m_2)=0$. For a wide-sense stationary process, the autocorrelation function is time-invariant and depends on the time difference $m= m_1–m_2$:

$$r_{xx}(m_1+\tau, m_2+\tau) = r_{xx}(m_1,m_2) = r_{xx}(m_1-m_2) = r_{xx}(m) \tag{3.37}$$

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

$$r_{xx}(-m) = r_{xx}(m) \tag{3.38}$$

$$r_{xx}(m) \leq r_{xx}(0) \tag{3.39}$$

Note that for a zero-mean signal, $r_{xx}(0)$ is the signal power.

Example 3.5 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

$$y(m)=\sum_k h_k x(m-k) \tag{3.40}$$

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

$$\begin{aligned} r_{yy}(k) &= \mathcal{E}[y(m)y(m+k)] \\ &= \sum_i \sum_j h_i h_j \mathcal{E}[x(m-i)x(m+k-j)] \\ &= \sum_i \sum_j h_i h_j r_{xx}(k+i-j) \end{aligned} \tag{3.41}$$

When the input $x(m)$ is an uncorrelated random signal with a unit variance, Equation (3.41) becomes

$$r_{yy}(k)=\sum_i h_i h_{k+i} \tag{3.42}$$

3.4.3 Autocovariance

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

$$\begin{aligned} c_{xx}(m_1,m_2) &= \mathcal{E}[(x(m_1)-\mu_x(m_1))(x(m_2)-\mu_x(m_2))] \\ &= r_{xx}(m_1,m_2) - \mu_x(m_1)\mu_x(m_2) \end{aligned} \tag{3.43}$$

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 (3.43) becomes

$$c_{xx}(m_1,m_2) = c_{xx}(m_1-m_2) = r_{xx}(m_1-m_2) - \mu_x^2 \tag{3.44}$$

3.4.4 Power Spectral Density

The power spectral density (PSD) function, also called the power spectrum, of a random process gives the spectrum of the distribution of the power among the individual frequency contents of the process. The power spectrum of a wide sense stationary process $X(m)$ is defined, by the Wiener–Khinchin theorem in Chapter 9, as the Fourier transform of the autocorrelation function:

$$\begin{aligned} P_{XX}(f) &= \mathcal{E}[X(f)X^*(f)] \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(k)e^{-j2\pi fm} \end{aligned} \tag{3.45}$$

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

$$P_{XX}(f) = r_{xx}(0) + \sum_{m=1}^{\infty} 2r_{xx}(m)\cos(2\pi fm) \tag{3.46}$$

The power spectral density is a real-valued non-negative function, expressed in units of watts per hertz. From Equation (3.45), the autocorrelation sequence of a random process may be obtained as the inverse Fourier transform of the power spectrum as

$$r_{xx}(m) = \int_{-1/2}^{1/2} P_{XX}(f)\, e^{j2\pi fm}\, df \tag{3.47}$$

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

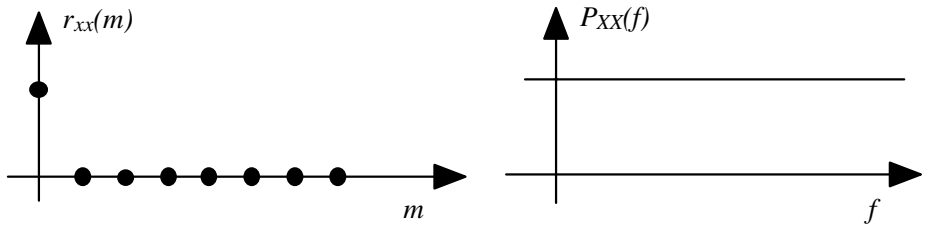


Figure 3.6 Autocorrelation and power spectrum of white noise.

Example 3.6 Power spectrum and autocorrelation of white noise (Figure3.6). 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:

$$r_{nn}(k) = \mathcal{E}[n(m)n(m+k)] = \begin{cases} \text{Noisepower} & k = 0 \\ 0 & k \neq 0 \end{cases} \tag{3.48}$$

Equation (3.48) 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 $r_{nn}(k)$:

$$P_{NN}(f) = \sum_{k=-\infty}^{\infty} r_{nn}(k)e^{-j2\pi fk} = r_{nn}(0) = \text{noise power} \tag{3.49}$$

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.

Example 3.7 Autocorrelation and power spectrum of impulsive noise. Impulsive noise is a random, binary-state ("on/off") sequence of impulses of random amplitudes and random time of occurrence. In Chapter 12, a random impulsive noise sequence $n_i(m)$ is modelled as an amplitude-modulated random binary sequence as

$$n_i(m) = n(m)b(m) \tag{3.50}$$

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

$$r_{nn}(k,m) = \mathcal{E}[n_i(m)n_i(m+k)] = \sigma_n^2\,\delta(k)b(m) \tag{3.51}$$

where σ_n^2 is the noise variance. Note that in Equation (3.51), the autocorrelation is expressed as a binary-state function that depends on the on/off state of impulsive noise at time *m*. The power spectrum of an impulsive noise sequence is obtained by taking the Fourier transform of the autocorrelation function:

$$P_{NN}(f,m) = \sigma_n^2\, b(m) \tag{3.52}$$

3.4.5 Joint Statistical Averages of Two Random Processes

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

$$f_{X(m_1)\cdots X(m_M),Y(n_1)\cdots Y(n_N)}(x_1,\ldots,x_M,y_1,\ldots,y_N) \tag{3.53}$$

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

$$\begin{aligned} &f_{X(m_1)\cdots X(m_M),Y(n_1)\cdots Y(n_N)}(x_1,\ldots,x_M,y_1,\ldots,y_N) \\ &\quad = f_{X(m_1)\cdots X(m_M)}(x_1,\ldots,x_M)\, f_{Y(n_1)\cdots Y(n_N)}(y_1,\ldots,y_N) \end{aligned} \tag{3.54}$$

3.4.6 Cross-Correlation and Cross-Covariance

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

$$\begin{aligned} r_{xy}(m_1,m_2) &= \mathcal{E}[x(m_1)y(m_2)] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(m_1)y(m_2)f_{X(m_1)Y(m_2)}(x(m_1),y(m_2))\,dx(m_1)\,dy(m_2) \end{aligned} \tag{3.55}$$

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

$$r_{xy}(m_1+\tau,m_2+\tau)=r_{xy}(m_1,m_2)=r_{xy}(m_1-m_2)=r_{xy}(m) \tag{3.56}$$

The cross-covariance function is defined as

$$\begin{aligned} c_{xy}(m_1,m_2) &= \mathcal{E}\left[\left(x(m_1)-\mu_x(m_1)\right)\left(y(m_2)-\mu_y(m_2)\right)\right] \\ &= r_{xy}(m_1,m_2)-\mu_x(m_1)\mu_y(m_2) \end{aligned} \tag{3.57}$$

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 (3.57) becomes

$$c_{xy}(m_1,m_2)=c_{xy}(m_1-m_2)=r_{xy}(m_1-m_2)-\mu_x\mu_y \tag{3.58}$$

Example 3.8 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

$$y_1(m)=x(m)+n_1(m) \tag{3.59}$$

$$y_2(m)=A\,x(m-D)+n_2(m) \tag{3.60}$$

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

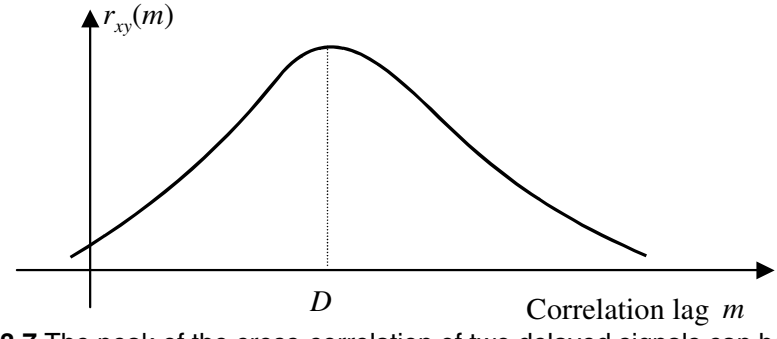


Figure 3.7 The peak of the cross-correlation of two delayed signals can be used to estimate the time delay D.

$$\begin{aligned} r_{y_1y_2}(k) &= \mathcal{E}[y_1(m)y_2(m+k)] \\ &= \mathcal{E}\{[x(m)+n_1(m)][Ax(m-D+k)+n_2(m+k)]\} \\ &= Ar_{xx}(k-D)+r_{xn_2}(k)+Ar_{xn_1}(k-D)+r_{n_1n_2}(k) \end{aligned} \tag{3.61}$$

Assuming that the signal and noise are uncorrelated, we have $r_{y_1y_2}(k) = Ar_{xx}(k-D)$. As shown in Figure 3.7, the cross-correlation function has its maximum at the lag D.

3.4.7 Cross-Power Spectral Density and Coherence

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

$$\begin{aligned} P_{XY}(f) &= \mathcal{E}[X(f)Y^*(f)] \\ &= \sum_{m=-\infty}^{\infty} r_{xy}(m)e^{-j2\pi fm} \end{aligned} \tag{3.62}$$

Like the cross-correlation the 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

$$C_{XY}(f) = \frac{P_{XY}(f)}{\sqrt{P_{XX}(f)P_{YY}(f)}} \tag{3.63}$$

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

3.4.8 Ergodic Processes and Time-Averaged Statistics

In many signal processing problems, there is only a single realisation of a random process from which its statistical parameters, such as the mean, the correlation and the power spectrum can be estimated. In such cases, time-averaged statistics, obtained from averages along the time dimension of a single realisation of the process, are used instead of the "true" 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

$$\underset{\text{along time } m}{\textit{statistical averages}}[x(m,s)] = \underset{\text{across space } s}{\textit{statistical averages}}[x(m,s)] \tag{3.64}$$

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

3.4.9 Mean-Ergodic Processes

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

$$\hat{\mu}_X = \frac{1}{N}\sum_{m=0}^{N-1} x(m) \tag{3.65}$$

A stationary process is said to be 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

$$\lim_{N\to\infty} \mathcal{E}[\hat{\mu}_X] = \mu_X \tag{3.66}$$

$$\lim_{N\to\infty} \text{var}[\hat{\mu}_X] = 0 \tag{3.67}$$

where μ_x is the "true" ensemble average of the process. Condition (3.67) is also referred to as mean-ergodicity in the mean square error (or minimum variance of error) sense. The time-averaged estimate of the mean of a signal, obtained from a random realisation of the process, is itself a random variable, with 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

$$\mathcal{E}[\hat{\mu}_x]=\mathcal{E}\left[\frac{1}{N}\sum_{m=0}^{N-1}x(m)\right]=\frac{1}{N}\sum_{m=0}^{N-1}\mathcal{E}[x(m)]=\frac{1}{N}\sum_{m=0}^{N-1}\mu_x=\mu_x \tag{3.68}$$

From Equation (3.68), the time-averaged estimate of the mean is unbiased. The variance of $\hat{\mu}_X$ is given by

$$\begin{aligned}\text{Var}[\hat{\mu}_x]&=\mathcal{E}[\hat{\mu}_x^2]-\mathcal{E}^2[\hat{\mu}_x]\\&=\mathcal{E}[\hat{\mu}_x^2]-\mu_x^2\end{aligned} \tag{3.69}$$

Now the term $\mathcal{E}[\hat{\mu}_x^2]$ in Equation (3.69) may be expressed as

$$\begin{aligned}\mathcal{E}[\hat{\mu}_x^2]&=\mathcal{E}\left[\left(\frac{1}{N}\sum_{m=0}^{N-1}x(m)\right)\left(\frac{1}{N}\sum_{k=0}^{N-1}x(k)\right)\right]\\&=\frac{1}{N}\sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)r_{xx}(m)\end{aligned} \tag{3.70}$$

Substitution of Equation (3.70) in Equation (3.69) yields

$$\begin{aligned}\text{Var}[\hat{\mu}_x^2]&=\frac{1}{N}\sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)r_{xx}(m)-\mu_x^2\\&=\frac{1}{N}\sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)c_{xx}(m)\end{aligned} \tag{3.71}$$

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

$$\lim_{N\to\infty}\frac{1}{N}\sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)c_{xx}(m)=0 \tag{3.72}$$

3.4.10 Correlation-Ergodic Processes

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

$$\hat{r}_{xx}(m) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)x(k+m) \tag{3.73}$$

A process is correlation-ergodic, in the mean square error sense, if

$$\lim_{N\to\infty} \mathcal{E}[\hat{r}_{xx}(m)] = r_{xx}(m) \tag{3.74}$$

$$\lim_{N\to\infty} \mathrm{Var}[\hat{r}_{xx}(m)] = 0 \tag{3.75}$$

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

$$\mathcal{E}[\hat{r}_{xx}(m)] = \mathcal{E}\left[\frac{1}{N}\sum_{k=0}^{N-1} x(k)x(k+m)\right] = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{E}[x(k)x(k+m)] = r_{xx}(m) \tag{3.76}$$

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

$$\mathrm{Var}[\hat{r}_{xx}(m)] = \mathcal{E}[\hat{r}_{xx}^2(m)] - r_{xx}^2(m) \tag{3.77}$$

The term $\mathcal{E}[\hat{r}_{xx}^2(m)]$ in Equation (3.77) may be expressed as

$$\begin{aligned}\mathcal{E}[\hat{r}_{xx}^2(m)] &= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1} \mathcal{E}[x(k)x(k+m)x(j)x(j+m)] \\ &= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1} \mathcal{E}[z(k,m)z(j,m)] \\ &= \frac{1}{N}\sum_{k=-N+1}^{N-1}\left(1-\frac{|k|}{N}\right) r_{zz}(k,m)\end{aligned} \tag{3.78}$$

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

$$\lim_{N\to\infty}\left[\frac{1}{N}\sum_{k=-N+1}^{N-1}\left(1-\frac{|k|}{N}\right) r_{zz}(k,m) - r_{xx}^2(m)\right] = 0 \tag{3.79}$$

3.5 Some Useful Classes of Random Processes

In this section, we consider some important classes of random processes extensively used in signal processing applications for the modelling of signals and noise.

3.5.1 Gaussian (Normal) Process

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

(a) Gaussian pdfs can model the distribution of many processes including some important classes of signals and noise.
(b) 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.
(c) Optimal estimation methods based on Gaussian models often result in linear and mathematically tractable solutions.
(d) The sum of many independent random processes has a Gaussian distribution. This is known as the central limit theorem.

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

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right] \tag{3.80}$$

where μ_x and σ_x^2 are the mean and the variance of the random variable x. The Gaussian process of Equation (3.80) is also denoted by $\mathcal{N}(x, \mu_x, \sigma_x^2)$. The maximum of a Gaussian pdf occurs at the mean μ_x, and is given by

$$f_X(\mu_x) = \frac{1}{\sqrt{2\pi}\,\sigma_x} \tag{3.81}$$

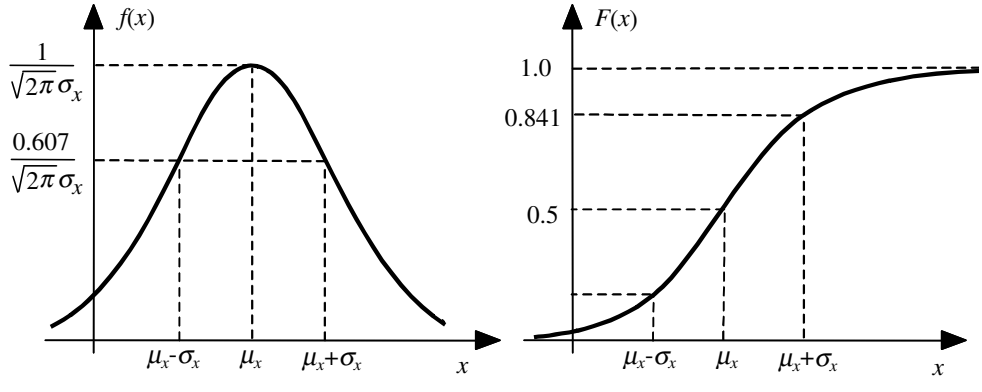


Figure 3.8 Gaussian probability density and cumulative density functions.

From Equation (3.80), the Gaussian pdf of x decreases exponentially with the increasing distance of x from the mean value μ_x. The distribution function $F(x)$ is given by

$$F_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \int_{-\infty}^{x} \exp\left(-\frac{(\chi - \mu_x)^2}{2\sigma_x^2}\right) d\chi \tag{3.82}$$

Figure 3.8 shows the pdf and the cdf of a Gaussian model.

3.5.2 Multivariate Gaussian Process

Multivariate 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

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}})\right] \tag{3.83}$$

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

$$\boldsymbol{\mu}_{\boldsymbol{x}} = \begin{pmatrix} \mathcal{E}[x(m_0)] \\ \mathcal{E}[x(m_2)] \\ \vdots \\ \mathcal{E}[x(m_{P-1})] \end{pmatrix} \tag{3.84}$$

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

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}} = \begin{pmatrix} c_{xx}(m_0, m_0) & c_{xx}(m_0, m_1) & \dots & c_{xx}(m_0, m_{P-1}) \\ c_{xx}(m_1, m_0) & c_{xx}(m_1, m_1) & \dots & c_{xx}(m_1, m_{P-1}) \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}(m_{P-1}, m_0) & c_{xx}(m_{P-1}, m_1) & \dots & c_{xx}(m_{P-1}, m_{P-1}) \end{pmatrix} \tag{3.85}$$

The Gaussian process of Equation (3.83) is also denoted by $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}})$. 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:

$$f_{\boldsymbol{X}}\left(\boldsymbol{x} = [x(m_0), \dots, x(m_{P-1})]^{\mathrm{T}}\right) = \prod_{i=0}^{P-1} \frac{1}{\sqrt{2\pi}\sigma_{xi}} \exp\left\{-\frac{[x(m_i) - \mu_{xi}]^2}{2\sigma_{xi}^2}\right\} \tag{3.86}$$

Example 3.9 Conditional multivariate Gaussian probability density function. Consider two vector realisations $\boldsymbol{x}(m)$ and $\boldsymbol{y}(m+k)$ from two vector-valued correlated stationary Gaussian processes $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{xx}})$ and $\mathcal{N}(\boldsymbol{y}, \boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{yy}})$. The joint probability density function of $\boldsymbol{x}(m)$ and $\boldsymbol{y}(m+k)$ is a multivariate Gaussian density $\mathcal{N}([\boldsymbol{x}(m), \boldsymbol{y}(m+k)], \boldsymbol{\mu}_{(\boldsymbol{x},\boldsymbol{y})}, \boldsymbol{\Sigma}_{(\boldsymbol{x},\boldsymbol{y})})$, with mean vector and covariance matrix given by

$$\boldsymbol{\mu}_{(\boldsymbol{x},\boldsymbol{y})} = \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{x}} \\ \boldsymbol{\mu}_{\boldsymbol{y}} \end{bmatrix} \tag{3.87}$$

$$\boldsymbol{\Sigma}_{(\boldsymbol{x},\boldsymbol{y})} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{xx}} & \boldsymbol{\Sigma}_{\boldsymbol{xy}} \\ \boldsymbol{\Sigma}_{\boldsymbol{yx}} & \boldsymbol{\Sigma}_{\boldsymbol{yy}} \end{bmatrix} \tag{3.88}$$

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

$$f_{\boldsymbol{X}|\boldsymbol{Y}}\left(\boldsymbol{x}(m)\middle|\boldsymbol{y}(m+k)\right)=\frac{f_{\boldsymbol{X},\boldsymbol{Y}}\left(\boldsymbol{x}(m),\boldsymbol{y}(m+k)\right)}{f_{\boldsymbol{Y}}\left(\boldsymbol{y}(m+k)\right)} \tag{3.89}$$

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

$$\begin{aligned}\boldsymbol{\mu}_{(\boldsymbol{x}|\boldsymbol{y})}&=\mathcal{E}\left[\boldsymbol{x}(m)\middle|\boldsymbol{y}(m+k)\right]\\&=\boldsymbol{\mu}_{\boldsymbol{x}}+\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_{\boldsymbol{y}})\end{aligned} \tag{3.90}$$

$$\boldsymbol{\Sigma}_{(\boldsymbol{x}|\boldsymbol{y})}=\boldsymbol{\Sigma}_{\boldsymbol{xx}}-\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{yx}} \tag{3.91}$$

3.5.3 Mixture Gaussian Process

Probability density functions of many 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. An M-mixture Gaussian density is defined as

$$f_{\boldsymbol{X}}(\boldsymbol{x})\ =\sum_{i=1}^{M}P_i\,\mathcal{N}_i(\boldsymbol{x},\boldsymbol{\mu}_{\boldsymbol{x}i},\boldsymbol{\Sigma}_{\boldsymbol{xx}i}) \tag{3.92}$$

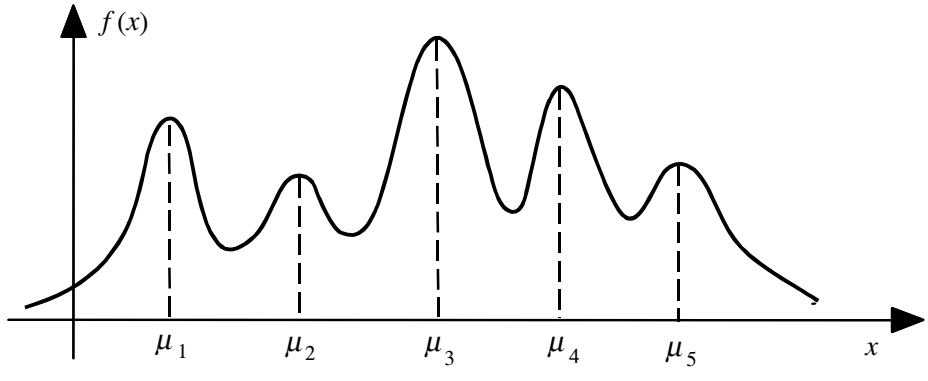


Figure 3.9 A mixture Gaussian pdf.

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

$$P_i = \frac{N_i}{\sum_{j=1}^{M} N_j} \tag{3.93}$$

where N_i is the number of observations associated with the mixture i. Figure 3.9 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.

3.5.4 A Binary-State Gaussian Process

Consider a random process $x(m)$ with two statistical states: 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 $\sigma^2_{x,1}$ (Figure 3.10). The state-dependent pdf of $x(m)$ can be expressed as

$$f_{X|S}(x(m)|s_i) = \frac{1}{\sqrt{2\pi}\sigma_{x,i}} \exp\left\{-\frac{1}{2\sigma^2_{x,i}}[x(m) - \mu_{x,i}]^2\right\}, \quad i=0, 1 \tag{3.94}$$

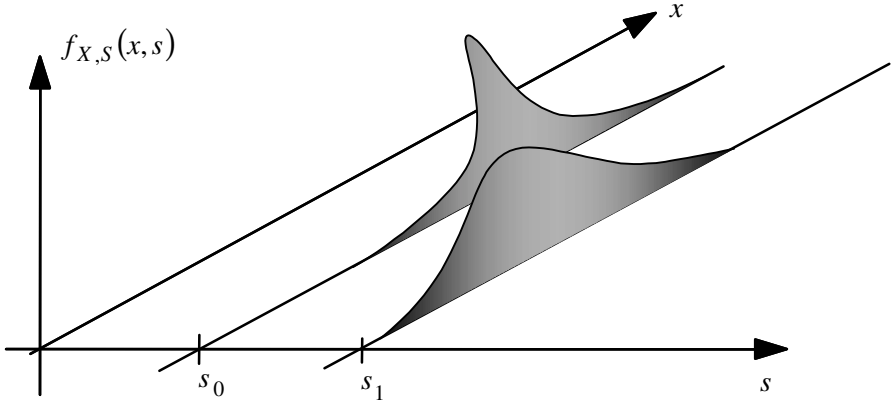


Figure 3.10 Illustration of a binary-state Gaussian process

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

$$\begin{aligned} f_{X,S}(x(m), s_i) &= f_{X|S}(x(m)|s_i) P_S(s_i) \\ &= \frac{1}{\sqrt{2\pi}\sigma_{x,i}} \exp\left\{ -\frac{1}{2\sigma_{x,i}^2} [x(m) - \mu_{x,i}]^2 \right\} P_S(s_i) \end{aligned} \tag{3.95}$$

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

$$\sum_S f_{X,S}(x(m), s_i) = f_X(x(m)) \tag{3.96}$$

$$\int_X f_{X,S}(x(m), s_i)\, dx = P_S(s_i) \tag{3.97}$$

and

$$\sum_S \int_X f_{X,S}(x(m), s_i)\, dx = 1 \tag{3.98}$$

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

3.5.5 Poisson Process

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

event is governed by a rate function $\lambda(t)$, such that the probability that an event occurs in a small time interval Δt is

$$Prob(1 \text{ occurrence in the interval } (t, t+\Delta t)) = \lambda(t)\Delta t \tag{3.99}$$

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

$$Prob(0 \text{ occurrence in the interval } (t, t+\Delta t)) = 1 - \lambda(t)\Delta t \tag{3.100}$$

when the parameter $\lambda(t)$ is independent of time, $\lambda(t)=\lambda$, and 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 $P(k, (0, t+\Delta t))$:

$$\begin{aligned} P(k,(0,t+\Delta t)) &= P(k,(0,t))P(0,(t,t+\Delta t)) + P(k-1,(0,t))P(1,(t,t+\Delta t)) \\ &= P(k,(0,t))(1-\lambda\Delta t) + P(k-1,(0,t))\lambda\Delta t \end{aligned} \tag{3.101}$$

Rearranging Equation (3.101), and letting Δt tend to zero, we obtain the following linear differential equation:

$$\frac{dP(k,t)}{dt} = -\lambda P(k,t) + \lambda P(k-1,t) \tag{3.102}$$

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

$$P(k,t) = \lambda e^{-\lambda t} \int_0^t P(k-1,\tau) e^{\lambda\tau}\, d\tau \tag{3.103}$$

Equation (3.103) 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

$$P(k,t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \tag{3.104}$$

From Equation (3.104), it is easy to show that for a homogenous Poisson process, the probability of k occurrences of an event in a time interval (t_1, t_2) is given by

$$P[k,(t_1,t_2)] = \frac{[\lambda(t_2 - t_1)]^k}{k!} e^{-\lambda(t_2-t_1)} \tag{3.105}$$

A Poisson counting process $X(t)$ is incremented by one every time the event occurs. From Equation (3.104), the mean and variance of a Poisson counting process $X(t)$ are

$$\mathcal{E}[X(t)] = \lambda t \tag{3.106}$$

$$r_{XX}(t_1,t_2) = \mathcal{E}[X(t_1)X(t_2)] = \lambda^2 t_1 t_2 + \lambda \min(t_1,t_2) \tag{3.107}$$

$$\text{Var}[X(t)] = \mathcal{E}\left[X^2(t)\right] - \mathcal{E}^2[X(t)] = \lambda t \tag{3.108}$$

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

3.5.6 Shot Noise

Shot noise happens when there is randomness in a directional flow of particles: as 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. The pulse sequence can be modelled as the response of a linear filter excited by a Poisson-distributed binary impulse input sequence (Figure 3.11). 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 have

$$\mathcal{E}[x(m\Delta t)] = 1 \times P(x(m\Delta t) = 1) + 0 \times P(x(m\Delta t) = 0) = \lambda \Delta t \tag{3.109}$$

and

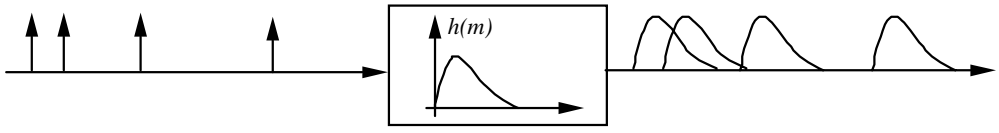


Figure 3.11 Shot noise is modelled as the output of a filter excited with a process.

$$\mathcal{E}[x(m\Delta t)x(n\Delta t)] = \begin{cases} 1\times P(x(m\Delta t)=1) = \lambda\Delta t, & m = n \\ 1\times P(x(m\Delta t)=1))\times P(x(n\Delta t)=1) = (\lambda\Delta t)^2, & m \neq n \end{cases} \tag{3.110}$$

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

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \\ &= \sum_{k=-\infty}^{\infty} x(m\Delta t)h(t-m\Delta t) \end{aligned} \tag{3.111}$$

where the binary signal $x(m\Delta t)$ can assume a value of 0 or 1. In Equation (3.111) 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

$$\begin{aligned} \mathcal{E}[y(t)] &= \sum_{k=-\infty}^{\infty} \mathcal{E}[x(m\Delta t)]h(t-m\Delta t) \\ &= \sum_{k=-\infty}^{\infty} \lambda\Delta t\, h(t-m\Delta t) \end{aligned} \tag{3.112}$$

and

$$\begin{aligned} r_{yy}(t_1,t_2) &= \mathcal{E}[y(t_1)\,y(t_2)] \\ &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \mathcal{E}[x(m\Delta t)x(n\Delta t)]h(t_1-n\Delta t)h(t_2-m\Delta t) \end{aligned} \tag{3.113}$$

Using Equation (3.110), the autocorrelation of $y(t)$ can be obtained as

$$r_{yy}(t_1,t_2)=\sum_{m=-\infty}^{\infty}(\lambda\Delta t)h(t_1-m\Delta t)h(t_2-m\Delta t)+\sum_{m=-\infty}^{\infty}\sum_{\substack{n=-\infty\\ n\neq m}}^{\infty}(\lambda\Delta t)^2\,h(t_1-m\Delta t)h(t_2-n\Delta t) \tag{3.114}$$

3.5.7 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 be used to model the random amplitude of the impulses. Finally, the finite duration character of real impulsive noise may be modelled by the impulse response of linear filter. The Poisson–Gaussian impulsive noise model is given by

$$x(m)=\sum_{k=-\infty}^{\infty}A_k h(m-\tau_k) \tag{3.115}$$

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^2 and τ_k is 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 radar target.

3.5.8 Markov Processes

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

$$\begin{aligned} f_X\left(x(m)=x_m \,|\, x(m-1)=x_{m-1},\ldots,x(m-N)=x_{m-N}\right) \\ = f_X\left(x(m)=x_m \,|\, x(m-1)=x_{m-1}\right) \end{aligned} \tag{3.116}$$

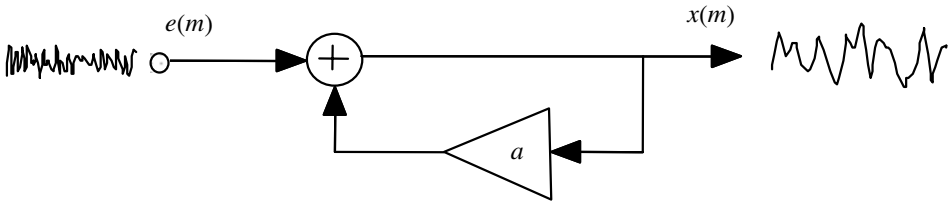


Figure 3.12 A first order autoregressive (Markov) process.

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

$$f_X(x(m)=x_m)=\int_{-\infty}^{\infty} f_X\left(x(m)=x_m|x(m-1)=x_{m-1}\right)f_X\left(x(m-1)=x_{m-1}\right)dx_{m-1} \tag{3.117}$$

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 3.10 A simple example of a Markov process is a first-order autoregressive process (Figure 3.12) defined as

$$x(m)=ax(m-1)+e(m) \tag{3.118}$$

In Equation (3.118), $x(m)$ 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

$$\begin{aligned} f_X\left(x(m)|x(m-1),...,x(m-N)\right) &= f_X\left(x(m)|x(m-1)\right) \\ &= f_E\left(e(m)=x(m)-ax(m-1)\right) \end{aligned} \tag{3.119}$$

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

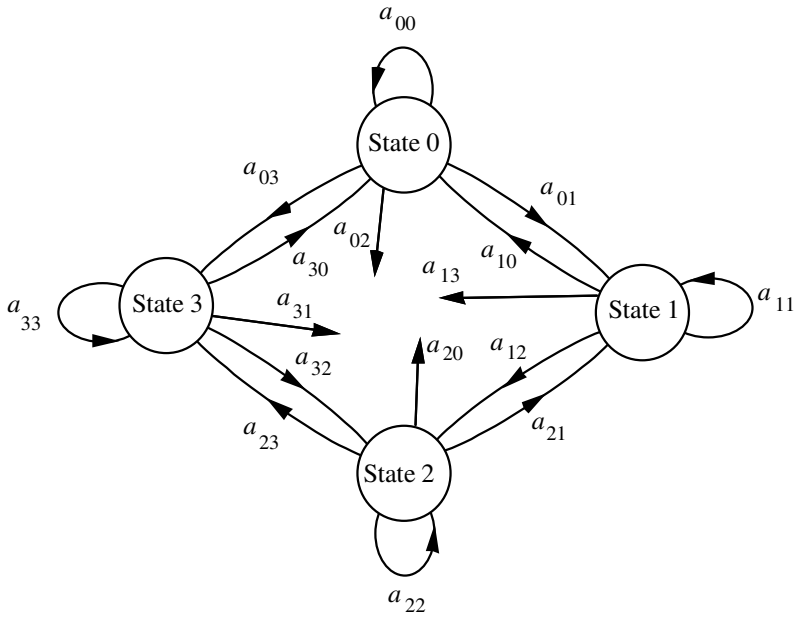


Figure 3.13 A Markov chain model of a four-state discrete-time Markov process.

$$\begin{aligned} f_X\left(x(m)|x(m-1)...,x(m-N)\right) &= f_X\left(x(m)|x(m-1)\right) \\ &= f_E\left(x(m)-ax(m-1)\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_e}\exp\left[-\frac{1}{2\sigma_e^2}\left(x(m)-ax(m-1)\right)^2\right] \end{aligned} \tag{3.120}$$

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

3.5.9 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 3.13). 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

$$a_{ij}(m-1,m)=Prob\big(x(m)=j|x(m-1)=i\big) \tag{3.121}$$

where $a_{ij}(m,m–1)$ 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 (3.121), the transition probability is expressed in a general time-dependent form. The marginal probability that a Markov process is in the state j at time m, $P_j(m)$, can be expressed as

$$P_j(m) = \sum_{i=1}^{N} P_i(m-1)\,a_{ij}(m-1,m) \tag{3.122}$$

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

number of states N
state probability vector

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

and the state transition matrix

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

Homogenous and Inhomogeneous Markov Chains

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

$$Prob\big(x(m)=j|x(m-1)=i\big)=a_{ij}(m-1,m)=a_{ij} \tag{3.123}$$

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

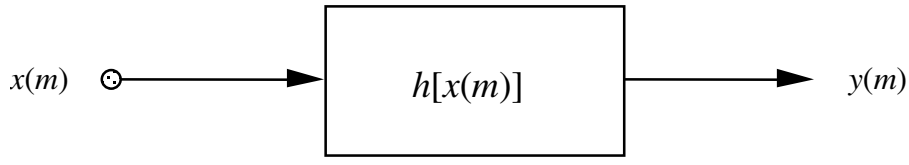


Figure 3.14 Transformation of a random process $x(m)$ to an output process $y(m)$.

3.6 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 3.14 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

$$P_Y\left(y(m) = y_j\right) = \sum_{x_i \to y_j} P_X\left(x(m) = x_i\right) \tag{3.124}$$

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

$$Prob[\,y(m)< Y < y(m)+\Delta y] = \int_{x(m)|\,y(m)<Y<y(m)+\Delta y} f_X\left(x(m)\right)dx(m) \tag{3.125}$$

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

3.6.1 Monotonic Transformation of Random Processes

Now for a monotonic one-to-one transformation $y(m)=h[x(m)]$ (e.g. as in Figure 3.15) Equation (3.125) becomes

$$Prob\left(y(m)< Y < y(m)+\Delta y\right) = Prob\left(x(m)< X < x(m)+\Delta x\right) \tag{3.126}$$

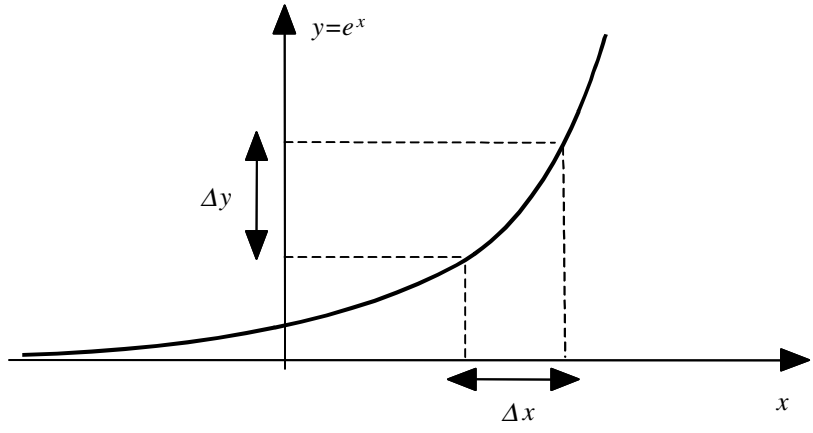


Figure 3.15 An example of a monotonic one-to-one mapping.

or, in terms of the cumulative distribution functions

$$F_Y(y(m)+\Delta y)-F_Y(y(m))=F_X(x(m)+\Delta x)-F_X(x(m)) \tag{3.127}$$

Multiplication of the left-hand side of Equation (3.127) by $\Delta y/\Delta y$ and the right-hand side by $\Delta x/\Delta x$ and re-arrangement of the terms yields

$$\frac{F_Y(y(m)+\Delta y)-F_Y(y(m))}{\Delta y}=\frac{\Delta x}{\Delta y}\,\frac{F_X(x(m)+\Delta x)-F_X(x(m))}{\Delta x} \tag{3.128}$$

Now as the intervals Δx and Δy tend to zero, Equation (3.128) becomes

$$f_Y(y(m))=\left|\frac{\partial x(m)}{\partial y(m)}\right| f_X(x(m)) \tag{3.129}$$

where $f_Y(y(m))$ is the probability density function. In Equation (3.129), substitution of $x(m)=h^{-1}(y(m))$ yields

$$f_Y(y(m))=\left|\frac{\partial\, h^{-1}(y(m))}{\partial\, y(m)}\right| f_X\left(h^{-1}(y(m))\right) \tag{3.130}$$

Equation (3.130) gives the pdf of the output signal in terms of the pdf of the input signal and the transformation.

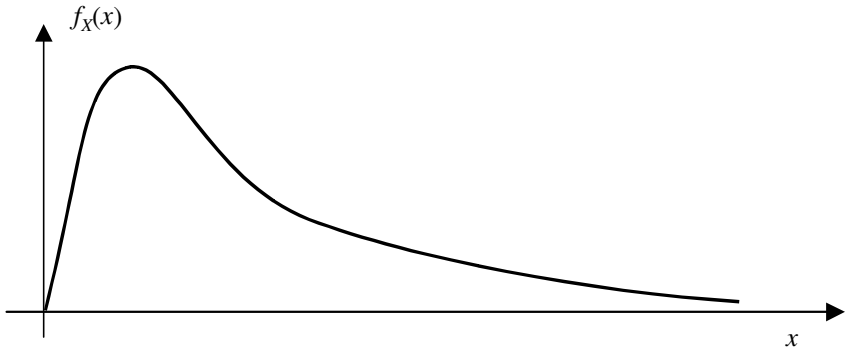


Figure 3.16 A log-normal distribution.

Example 3.11 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 as in Equation (3.80) then the non-negative valued variable $y(m)=\exp(x(m))$ has a log-normal distribution (Figure 3.16) obtained using Equation (3.130) as

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\,\sigma_x\, y(m)} \exp\left\{-\frac{[\ln y(m) - \mu_x]^2}{2\sigma_x^2}\right\} \tag{3.131}$$

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

$$\mu_x = \ln \mu_y - \frac{1}{2}\ln\left(1+\sigma_y^2/\mu_y^2\right) \tag{3.132}$$

$$\sigma_x^2 = \ln\left(1+\sigma_y^2/\mu_y^2\right) \tag{3.133}$$

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

$$\mu_y = \exp(\mu_x + \sigma_x^2/2) \tag{3.134}$$

$$\sigma_y^2 = \mu_x^2\,[\exp(\sigma_y^2) - 1] \tag{3.135}$$

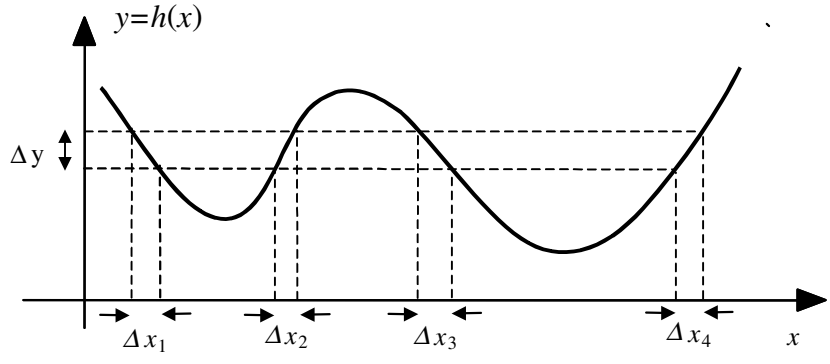


Figure 3.17 Illustration of a many to one transformation.

3.6.2 Many-to-One Mapping of Random Signals

Now consider the case when the transformation $h(\cdot)$ is a non-monotonic function such as that shown in Figure 3.17. 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

$$Prob(y(m) < Y < y(m)+\Delta y) = \sum_{k=1}^{K} Prob(x_k(m) < X < x_k(m)+\Delta x_k) \quad (3.136)$$

where x_k is the k^{th} root of $y(m)=h(x(m))$. Similar to the development in Section 3.6.1, Equation (3.136) can be written as

$$\frac{F_Y(y(m)+\Delta y)-F_Y(y(m))}{\Delta y}\Delta y = \sum_{k=1}^{K} \frac{F_X(x_k(m)+\Delta x_k)-F_X(x_k(m))}{\Delta x_k}\Delta x_k \quad (3.137)$$

Equation (3.137) can be rearranged as

$$\frac{F_Y(y(m)+\Delta y)-F_Y(y(m))}{\Delta y} = \sum_{k=1}^{K} \frac{\Delta x_k}{\Delta y}\frac{F_X(x_k(m)+\Delta x_k)-F_X(x_k(m))}{\Delta x_k} \quad (3.138)$$

Now as the intervals Δx and Δy tend to zero Equation (3.138) becomes

$$f_Y(y(m)) = \sum_{k=1}^{K} \left| \frac{\partial x_k(m)}{\partial y(m)} \right| f_X(x_k(m)) = \sum_{k=1}^{K} \frac{1}{|h'(x_k(m))|} f_X(x_k(m)) \tag{3.139}$$

where $h'(x_k(m)) = \partial h(x_k(m)) / \partial x_k(m)$. Note that for a monotonic function, $K=1$ and Equation (3.139) becomes the same as Equation (3.130). Equation (3.139) can be expressed as

$$f_Y(y(m)) = \sum_{k=1}^{K} |J(x_k(m))|^{-1} f_X(x_k(m)) \tag{3.140}$$

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

$$\boldsymbol{y}(m) = \boldsymbol{H}(\boldsymbol{x}(m)) \tag{3.141}$$

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

$$f_{\boldsymbol{Y}}(\boldsymbol{y}(m)) = \sum_{k=1}^{K} |\boldsymbol{J}(\boldsymbol{x}_k(m))|^{-1} f_{\boldsymbol{X}}(\boldsymbol{x}_k(m)) \tag{3.142}$$

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

$$|\boldsymbol{J}(\boldsymbol{x})| = \begin{vmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_P} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_P}{\partial x_1} & \frac{\partial y_P}{\partial x_2} & \cdots & \frac{\partial y_P}{\partial x_P} \end{vmatrix} \tag{3.143}$$

For a monotonic linear vector transformation such as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{3.144}$$

the pdf of $\boldsymbol{y}$ becomes

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = |\boldsymbol{J}|^{-1} f_{\boldsymbol{X}}(\boldsymbol{H}^{-1}\boldsymbol{y}) \tag{3.145}$$

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

Example 3.12 The input–output relation of a $P \times P$ linear transformation matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{y} = \boldsymbol{H}\,\boldsymbol{x} \tag{3.146}$$

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 $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ and a probability density function given by:

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \exp\left[-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{x}\right] \tag{3.147}$$

From Equations (3.145)–(3.147), the pdf of the output $\boldsymbol{y}$ is given by

$$\begin{aligned} f_{\boldsymbol{Y}}(\boldsymbol{y}) &= \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{H}^{-1^{\mathrm{T}}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{H}^{-1}\boldsymbol{y}\right)|\boldsymbol{H}|^{-1} \\ &= \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}|\boldsymbol{H}|} \exp\left(-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}\boldsymbol{y}\right) \end{aligned} \tag{3.148}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{yy}} = \boldsymbol{H}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{H}^{\mathrm{T}}$. Note that a linear transformation of a Gaussian process yields another Gaussian process.

3.7 Summary

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

its Fourier transform the power spectrum are particularly important because they can be used to identify the patterns in a stochastic process.

We considered the concepts of stationary, ergodic stationary and 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 with a chain of stationary subprocesses as described in Chapter 5 on hidden Markov models. For signal processing applications, a number of useful pdfs, including the Gaussian, the mixture Gaussian, the Markov and the Poisson process, were considered. These pdf models are extensively employed in the remainder of this book. Signal processing normally involves the filtering or transformation of an input signal to an output signal. We derived general expressions for the pdf of the output of a system in terms of the pdf of the input. We also considered some applications of stochastic processes for modelling random noise such as white noise, clutters, shot noise and impulsive noise.

Bibliography

ANDERSON O.D. (1976) Time Series Analysis and Forecasting. The Box–Jenkins Approach. Butterworth, London.

AYRE A.J. (1972) Probability and Evidence. Columbia University Press, New York.

BARTLETT M.S. (1960) Stochastic Processes. Cambridge University Press.

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. Wiley, New York.

CARTER G. (1987) Coherence and Time Delay Estimation. Proc. IEEE, **75**, **2**, pp. 236–55.

CLARK A.B. and DISNEY R.L. (1985) Probability and Random Processes, 2nd Ed. Wiley, 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.

GAUSS C.F. (1963) Theory of Motion of Heavenly Bodies. Dover, New York.

GARDENER W.A. (1986) Introduction to Random Processes: With Application to Signals and Systems. Macmillan, New York.

HELSTROM C.W. (1991) Probability and Stochastic Processes for Engineers. Macmillan, New York.

ISAACSON D. and MASDEN R. (1976) Markov Chains Theory and Applications. Wiley, New York.

JEFFREY H. (1961) Scientific Inference, 3rd Ed. Cambridge University Press.

JEFFREY H. (1973) Theory of Probability, 3rd Ed. Clarendon Press, Oxford.

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

KOLMOGOROV A.N. (1956) Foundations of the Theory of Probability. Chelsea Publishing Company, New York.

KENDALL M. and STUART A. (1977) The Advanced Theory of Statistics. Macmillan.

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**, pp. 153–162.

MEYER P.L. (1970) Introductory Probability and Statistical Applications. Addison-Wesley, Reading, MA.

PEEBLES P.Z. (1987) Probability, Random Variables and Random Signal Principles. McGraw-Hill, New York.

PARZEN E. (1962) Stochastic Processes. Holden-Day, San Francisco.

PAPOPULIS A. (1984) Probability, Random Variables and Stochastic Processes. McGraw-Hill, New York.

PAPOPULIS A. (1977) Signal Analysis, McGraw-Hill, New York.

RAO C.R. (1973) Linear Statistical Inference and Its Applications. Wiley, New York.

ROZANOV Y.A. (1969) Probability Theory: A Concise Course, Dover, New York.

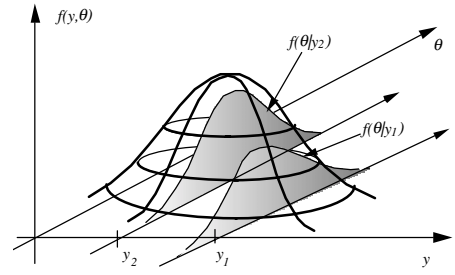
SHANMUGAN K.S. and BREIPOHL A.M. (1988) Random Signals: Detection, Estimation and Data Analysis. Wiley, New York.

THOMAS J.B. (1988) An introduction to Applied probability and Random Processes. Huntington, Krieger Publishing, New York.

WOZENCRAFT J.M. and JACOBS I.M. (1965) Principles of Communication Engineering. Wiley, New York

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)



4

BAYESIAN ESTIMATION

Bayesian estimation is a framework for the formulation of statistical inference problems. In the prediction or estimation of a random process from a related observation signal, the Bayesian philosophy is based on combining the evidence contained in the signal with prior knowledge of the probability distribution of the process. Bayesian methodology includes the classical estimators such as maximum a posteriori (MAP), maximum-likelihood (ML), minimum mean square error (MMSE) and minimum mean absolute value of error (MAVE) as special cases. The hidden Markov model, widely used in statistical signal processing, is an example of a Bayesian model. Bayesian inference is based on minimisation of the so-called Bayes' risk function, which includes a posterior model of the unknown parameters given the observation and a cost-of-error function. This chapter begins with an introduction to the basic concepts of estimation theory, and considers the statistical measures that are used to quantify the performance of an estimator. We study Bayesian estimation methods and consider the effect of using a prior model on the mean and the variance of an estimate. The estimate–maximise (EM) method for the estimation of a set of unknown parameters from an incomplete observation is studied, and applied to the mixture Gaussian modelling of the space of a continuous random variable. This chapter concludes with an introduction to the Bayesian classification of discrete or finite-state signals, and the K-means clustering method.

4.1 Bayesian Estimation Theory: Basic Definitions

Estimation theory is concerned with the determination of the best estimate of an unknown parameter vector from an observation signal, or the recovery of a clean signal degraded by noise and distortion. For example, given a noisy sine wave, we may be interested in estimating its basic parameters (i.e. amplitude, frequency and phase), or we may wish to recover the signal itself. An estimator takes as the input a set of noisy or incomplete observations, and, using a dynamic model (e.g. a linear predictive model) and/or a probabilistic model (e.g. Gaussian model) of the process, estimates the unknown parameters. The estimation accuracy depends on the available information and on the efficiency of the estimator. In this chapter, the Bayesian estimation of continuous-valued parameters is studied. The modelling and classification of finite-state parameters is covered in the next chapter.

Bayesian theory is a general inference framework. In the estimation or prediction of the state of a process, the Bayesian method employs both the evidence contained in the observation signal and the accumulated prior probability of the process. Consider the estimation of the value of a random parameter vector $\boldsymbol{\theta}$, given a related observation vector $\boldsymbol{y}$. From Bayes' rule the posterior probability density function (pdf) of the parameter vector $\boldsymbol{\theta}$ given $\boldsymbol{y}$, $f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})$, can be expressed as

$$f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \tag{4.1}$$

where for a given observation, $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is a constant and has only a normalising effect. Thus there are two variable terms in Equation (4.1): one term $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$ 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 $\boldsymbol{\theta}$. The relative influence of the likelihood pdf $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$ and the prior pdf $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ on the posterior pdf $f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y})$ depends on the shape of these function, 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 pdf will have no influence.

The remainder of this chapter is concerned with different forms of Bayesian estimation and its applications. First, in this section, some basic concepts of estimation theory are introduced.

4.1.1 Dynamic and Probability Models in Estimation

Optimal estimation algorithms utilise dynamic and statistical 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. A statistical probability model characterises the random fluctuations of a signal in terms of its statistics, such as the mean and the covariance, and most completely in terms of a probability model. Conditional probability models, in addition to modelling the random fluctuations of a signal, can also model the dependence of the signal on its past values or on some other related process.

As an illustration consider the estimation of a P-dimensional parameter vector $\boldsymbol{\theta}=[\theta_0, \theta_1, ..., \theta_{P-1}]$ from a noisy observation vector $\boldsymbol{y}=[y(0), y(1), ..., y(N-1)]$ modelled as

$$\boldsymbol{y}=h(\boldsymbol{\theta},\boldsymbol{x},\boldsymbol{e})+\boldsymbol{n} \tag{4.2}$$

where, as illustrated in Figure 4.1, the function $h(\cdot)$ with a random input $\boldsymbol{e}$, output $\boldsymbol{x}$, and parameter vector $\boldsymbol{\theta}$, is a predictive model of the signal $\boldsymbol{x}$, and $\boldsymbol{n}$ is an additive random noise process. In Figure 4.1, the distributions of the random noise $\boldsymbol{n}$, the random input $\boldsymbol{e}$ and the parameter vector $\boldsymbol{\theta}$ are modelled by probability density functions, $f_{\boldsymbol{N}}(\boldsymbol{n})$, $f_{\boldsymbol{E}}(\boldsymbol{e})$, and $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ 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 noisy observation. In general, the more modelling information used in an estimation process, the better the results, provided that the models are an accurate characterisation of the observation and the parameter process.

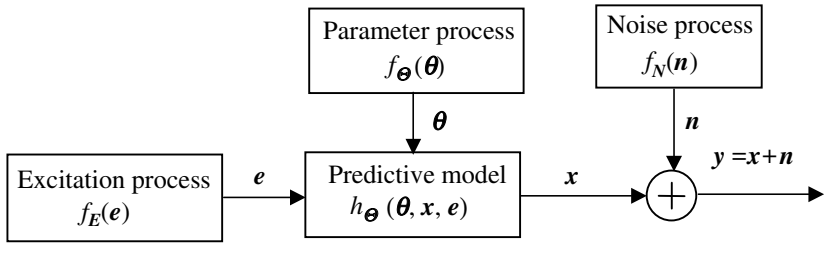


Figure 4.1 A random process $\boldsymbol{y}$ is described in terms of a predictive model $h(\cdot)$, and statistical models $f_{\boldsymbol{E}}(\cdot)$, $f_{\boldsymbol{\Theta}}(\cdot)$ and $f_{\boldsymbol{N}}(\cdot)$.

4.1.2 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 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 "character" (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. 4.2 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 are 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.

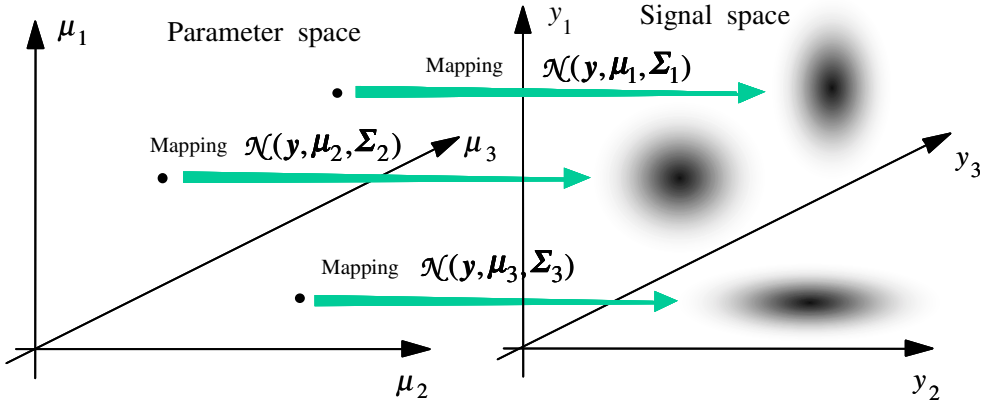


Figure 4.2 Illustration of three points in the parameter space of a Gaussian process and the associated signal spaces, for simplicity the variances are not shown in parameter space.

4.1.3 Parameter Estimation and Signal Restoration

Parameter estimation and signal restoration are closely related problems. The main difference is due to the rapid fluctuations of most signals in comparison with the relatively slow variations of most parameters. For example, speech sounds fluctuate at speeds of up to 20 kHz, whereas the underlying vocal tract and pitch parameters vary at a relatively lower rate of less than 100 Hz. This observation implies that normally more averaging can be done in parameter estimation than in signal restoration.

As a simple example, consider a signal observed in a zero-mean random noise process. Assume we wish to estimate (a) the average of the clean signal and (b) the clean signal itself. As the observation length increases, the estimate of the signal mean approaches the mean value of the clean signal, whereas the estimate of the clean signal samples depends on the correlation structure of the signal and the signal-to-noise ratio as well as on the estimation method used.

As a further example, consider the interpolation of a sequence of lost samples of a signal given N recorded samples, as illustrated in Figure 4.3. Assume that an autoregressive (AR) process is used to model the signal as

$$\boldsymbol{y} = \boldsymbol{X\theta} + \boldsymbol{e} + \boldsymbol{n} \tag{4.3}$$

where $\boldsymbol{y}$ is the observation signal, $\boldsymbol{X}$ is the 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

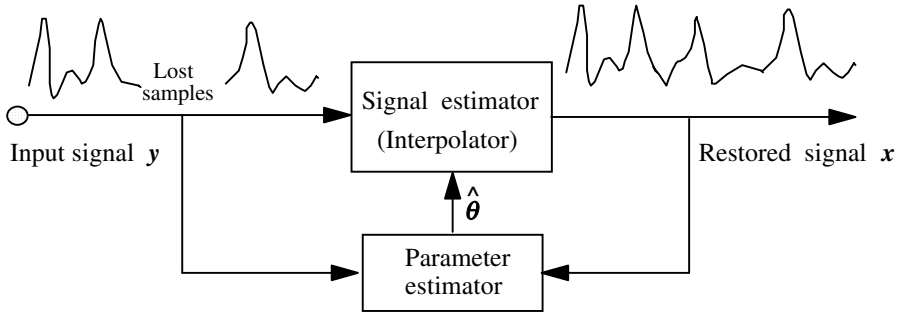


Figure 4.3 Illustration of signal restoration using a parametric model of the signal process.

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. In this chapter we are concerned with the parameter estimation problem, although the same ideas also apply to signal interpolation, which is considered in Chapter 11.

4.1.4 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

$$\hat{\boldsymbol{\theta}} = f(\boldsymbol{y}, N, \mathcal{M}) \tag{4.4}$$

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:

(a) *Expected value* of estimate: $\mathcal{E}[\hat{\boldsymbol{\theta}}]$

(b) *Bias* of estimate: $\mathcal{E}[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] = \mathcal{E}[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta}$

(c) *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:

(a) Unbiased estimator: an estimator of $\boldsymbol{\theta}$ is unbiased if the expectation of the estimate is equal to the true parameter value:

$$\mathcal{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta} \tag{4.5}$$

An estimator is *asymptotically unbiased* if for increasing length of observations N we have

$$\lim_{N \to \infty} \mathcal{E}[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta} \tag{4.6}$$

(b) 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 $\boldsymbol{\theta}$:

$$\text{Cov}[\hat{\boldsymbol{\theta}}_{\text{Efficient}}] \leq \text{Cov}[\hat{\boldsymbol{\theta}}] \tag{4.7}$$

where $\hat{\boldsymbol{\theta}}$ is any other estimate of $\boldsymbol{\theta}$.

(c) 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:

$$\lim_{N \to \infty} P[|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}| > \varepsilon] = 0 \tag{4.8}$$

where ε is arbitrary small.

Example 4.1 Consider the bias in the time-averaged estimates of the mean μ_y and the variance σ_y^2 of N observation samples $[y(0), ..., y(N-1)]$, of an ergodic random process, given as

$$\hat{\mu}_y = \frac{1}{N} \sum_{m=0}^{N-1} y(m) \tag{4.9}$$

$$\hat{\sigma}_y^2 = \frac{1}{N} \sum_{m=0}^{N-1} \left[y(m) - \hat{\mu}_y \right]^2 \tag{4.10}$$

It is easy to show that $\hat{\mu}_y$ is an unbiased estimate, since

$$\mathcal{E}[\hat{\mu}_y] = \frac{1}{N} \sum_{m=0}^{N-1} \mathcal{E}[y(m)] = \mu_y \tag{4.11}$$

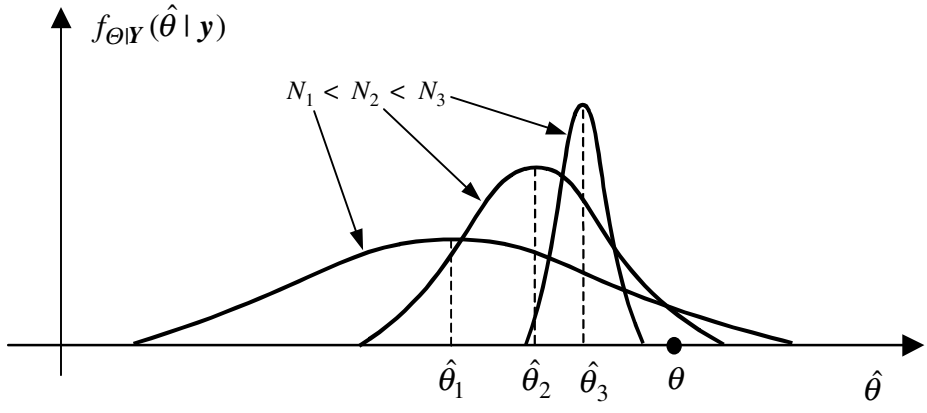


Figure 4.4 Illustration of the decrease in the bias and variance of an asymptotically unbiased estimate of the parameter θ with increasing length of observation.

The expectation of the estimate of the variance can be expressed as

$$\begin{aligned} \mathcal{E}\left[\hat{\sigma}_y^2\right] &= \mathcal{E}\left[\frac{1}{N}\sum_{m=0}^{N-1}\left(y(m)-\frac{1}{N}\sum_{k=0}^{N-1}y(k)\right)^2\right] \\ &= \sigma_y^2 - \frac{2}{N}\sigma_y^2 + \frac{1}{N}\sigma_y^2 \\ &= \sigma_y^2 - \frac{1}{N}\sigma_y^2 \end{aligned} \tag{4.12}$$

From Equation (4.12), the bias in the estimate of the variance is inversely proportional to the signal length N, and vanishes as N tends to infinity; hence the estimate is asymptotically unbiased. In general, the bias and the variance of an estimate decrease with increasing number of observation samples N and with improved modelling. Figure 4.4 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.

4.1.5 Prior and Posterior Spaces and Distributions

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. The *posterior signal* or *parameter space* is the subspace of all the likely values of a signal or a parameter consistent with *both* the prior information and the evidence in the *observation*. Consider a random process with a parameter

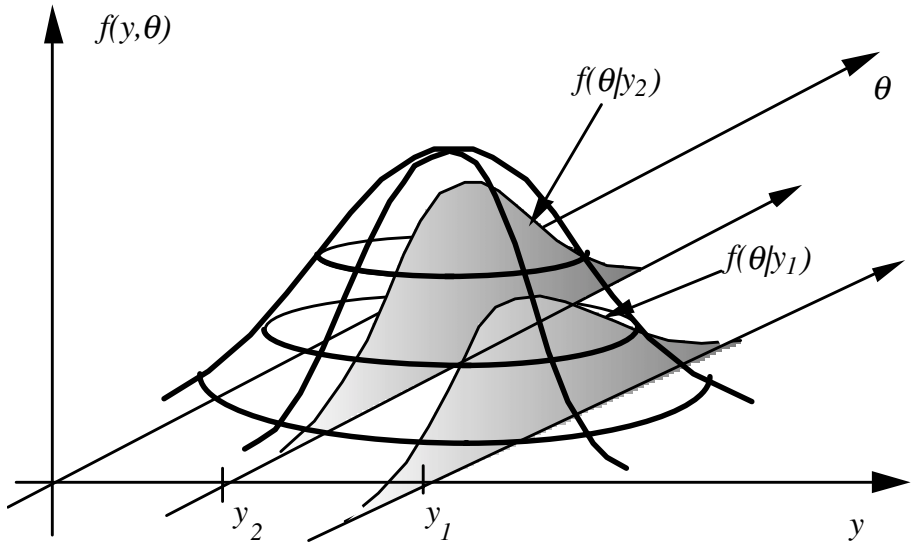


Figure 4.5 Illustration of joint distribution of signal y and parameter θ and the posterior distribution of θ given y.

space $\boldsymbol{\Theta}$ observation space $\boldsymbol{Y}$ and a joint pdf $f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})$. From the Bayes' rule the posterior pdf of the parameter vector $\boldsymbol{\theta}$, given an observation vector $\boldsymbol{y}$, $f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})$, can be expressed as

$$\begin{aligned} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \\ &= \frac{f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, d\boldsymbol{\theta}} \end{aligned} \tag{4.13}$$

where, for a given observation vector $\boldsymbol{y}$, the pdf $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is a constant and has only a normalising effect. From Equation (4.13), the posterior pdf is proportional to the product of the likelihood $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$ that the observation $\boldsymbol{y}$ was generated by the parameter vector $\boldsymbol{\theta}$, and the prior pdf $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. The prior pdf gives the unconditional parameter distribution *averaged* over the entire observation space as

$$f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \int_{\boldsymbol{Y}} f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})\, d\boldsymbol{y} \tag{4.14}$$

For most applications, it is relatively convenient to obtain the likelihood function $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$. The *prior* pdf *influences* the inference drawn from the likelihood function by weighting it with $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. 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 and the observation. Figure 4.5 illustrates the joint pdf of a signal $y(m)$ and a parameter θ. The prior pdf of θ can be obtained by integrating $f_{Y|\Theta}(y(m)|\theta)$ with respect to $y(m)$. As shown, an observation $y(m)$ cuts a posterior pdf $f_{\Theta|Y}(\theta|y(m))$ through the joint distribution.

Example 4.2 A noisy signal vector of length N samples is modelled as

$$\boldsymbol{y}(m)=\boldsymbol{x}(m)+\boldsymbol{n}(m) \tag{4.15}$$

Assume that the signal $\boldsymbol{x}(m)$ is Gaussian with mean vector $\boldsymbol{\mu}_{\boldsymbol{x}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$, and that the noise $\boldsymbol{n}(m)$ is also Gaussian with mean vector $\boldsymbol{\mu}_{\boldsymbol{n}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{nn}}$. 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 $\boldsymbol{y}(m) - \boldsymbol{\mu}_{\boldsymbol{n}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{nn}}$ as shown in Figure 4.6.The likelihood function is given by

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}(m)|\boldsymbol{x}(m)) &= f_{\boldsymbol{N}}(\boldsymbol{y}(m)-\boldsymbol{x}(m)) \\ &= \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_{\boldsymbol{nn}}|^{1/2}} \exp\left\{-\frac{1}{2}[\boldsymbol{x}(m)-(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{n}})]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{nn}}^{-1}[\boldsymbol{x}(m)-(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{n}})]\right\} \end{aligned} \tag{4.16}$$

where the terms in the exponential function have been rearranged to emphasize the illustration of the likelihood space in Figure 4.6. Hence the posterior pdf can be expressed as

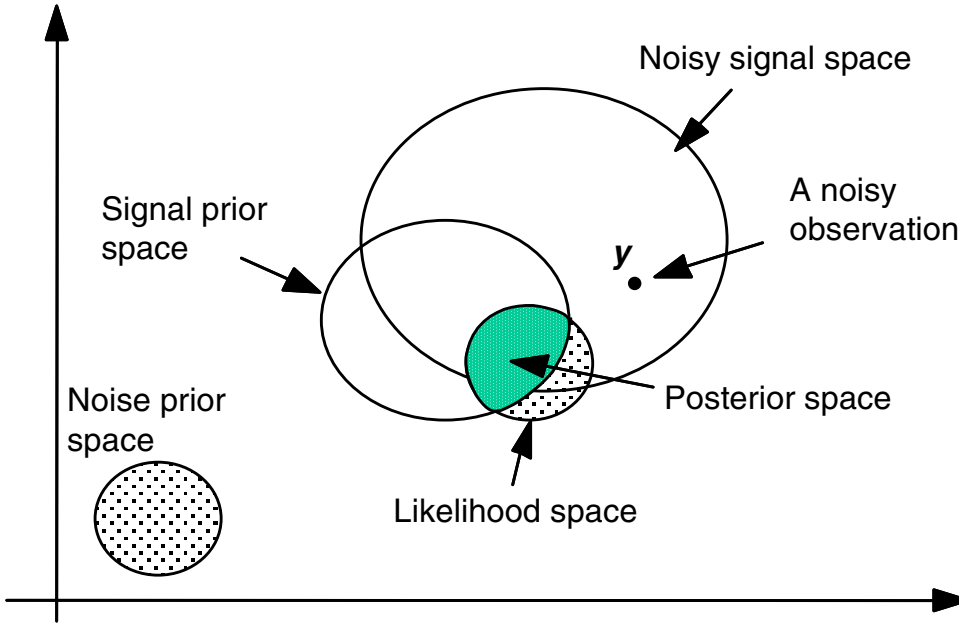


Figure 4.6 Sketch of a two-dimensional signal and noise spaces, and the likelihood and posterior spaces of a noisy observation $\boldsymbol{y}$.

$$f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}(m)|\boldsymbol{y}(m)) = \frac{f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}(m)|\boldsymbol{x}(m)) f_{\boldsymbol{X}}(\boldsymbol{x}(m))}{f_{\boldsymbol{Y}}(\boldsymbol{y}(m))}$$

$$= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y}(m))} \frac{1}{(2\pi)^N |\boldsymbol{\Sigma}_{\boldsymbol{nn}}|^{1/2} |\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}}$$

$$\times \exp\left(-\frac{1}{2}\left\{[\boldsymbol{x}(m)-(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{n}})]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{nn}}^{-1} [\boldsymbol{x}(m)-(\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{n}})] + (\boldsymbol{x}(m)-\boldsymbol{\mu}_{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1} (\boldsymbol{x}(m)-\boldsymbol{\mu}_{\boldsymbol{x}})\right\}\right) \tag{4.17}$$

For a two-dimensional signal and noise process, the prior spaces of the signal, the noise, and the noisy signal are illustrated in Figure 4.6. Also illustrated are the likelihood and posterior spaces for a noisy observation vector $\boldsymbol{y}$. Note that the centre of the posterior space is obtained by subtracting the noise mean vector from the noisy signal vector. The clean signal is then somewhere within a subspace determined by the noise variance.

4.2 Bayesian Estimation

The Bayesian estimation of a parameter vector $\boldsymbol{\theta}$ is based on the minimisation of a Bayesian risk function defined as an average cost-of-error function:

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{\theta}}) &= \mathcal{E}[C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})] \\ &= \int_{\boldsymbol{\theta}}\int_{\boldsymbol{Y}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})\, d\boldsymbol{y}\, d\boldsymbol{\theta} \\ &= \int_{\boldsymbol{\theta}}\int_{\boldsymbol{Y}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, f_{\boldsymbol{Y}}(\boldsymbol{y})\, d\boldsymbol{y}\, d\boldsymbol{\theta}\end{aligned} \tag{4.18}$$

where the cost-of-error function $C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})$ allows the appropriate weighting of the various outcomes to achieve desirable objective or subjective properties. The cost function can be chosen to associate a high cost with outcomes that are undesirable or disastrous. For a given observation vector $\boldsymbol{y}$, $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is a constant and has no effect on the risk-minimisation process. Hence Equation (4.18) may be written as a conditional risk function:

$$\mathcal{R}(\hat{\boldsymbol{\theta}}\,|\boldsymbol{y}) = \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.19}$$

The Bayesian estimate obtained as the minimum-risk parameter vector is given by

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\boldsymbol{\theta}}} \mathcal{R}(\hat{\boldsymbol{\theta}}\,|\,\boldsymbol{y}) = \arg\min_{\hat{\boldsymbol{\theta}}} \left[\int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\boldsymbol{y})\, d\boldsymbol{\theta} \right] \tag{4.20}$$

Using Bayes' rule, Equation (4.20) can be written as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\min_{\hat{\boldsymbol{\theta}}} \left[\int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})\, f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \right] \tag{4.21}$$

Assuming that the risk function is differentiable, and has a well-defined minimum, the Bayesian estimate can be obtained as

$$\hat{\boldsymbol{\theta}}_{\text{Bayesian}} = \arg\operatorname{zero}_{\hat{\boldsymbol{\theta}}} \frac{\partial \mathcal{R}(\hat{\boldsymbol{\theta}}\,|\,\boldsymbol{y})}{\partial \hat{\boldsymbol{\theta}}} = \arg\operatorname{zero}_{\hat{\boldsymbol{\theta}}} \left[\frac{\partial}{\partial \hat{\boldsymbol{\theta}}} \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}) f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, d\boldsymbol{\theta} \right] \tag{4.22}$$

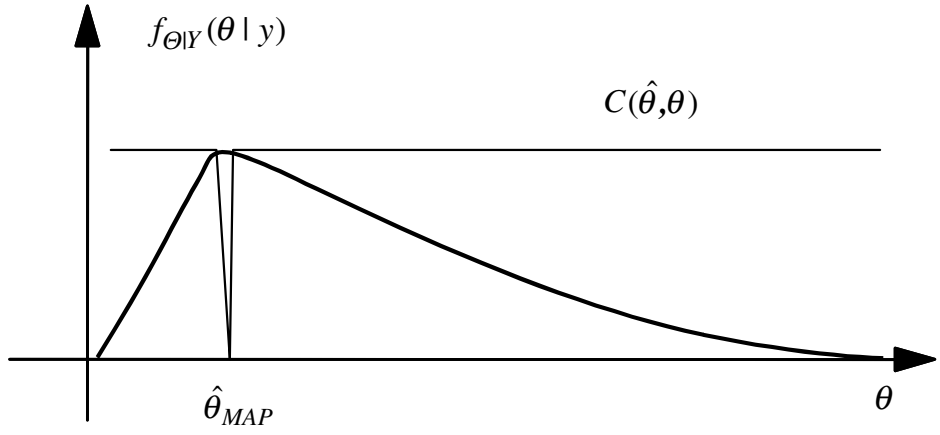


Figure 4.7 Illustration of the Bayesian cost function for the MAP estimate.

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 $f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y})$. The MAP estimate corresponds to a Bayesian estimate with a so-called uniform cost function (in fact, as shown in Figure 4.7 the cost function is notch-shaped) defined as

$$C(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}) = 1 - \delta(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}) \tag{4.23}$$

where $\delta(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})$ is the Kronecker delta function. Substitution of the cost function in the Bayesian risk equation yields

$$\begin{aligned} \mathcal{R}_{MAP}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) &= \int_{\boldsymbol{\theta}} [1-\delta(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})] f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \\ &= 1 - f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) \end{aligned} \tag{4.24}$$

From Equation (4.24), the minimum Bayesian risk estimate corresponds to the parameter value where the posterior function attains a maximum. Hence the MAP estimate of the parameter vector $\boldsymbol{\theta}$ is obtained from a minimisation of the risk Equation (4.24) or equivalently maximisation of the posterior function:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{MAP} &= \arg\max_{\boldsymbol{\theta}} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{\theta}} [f_{\boldsymbol{Y}|\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})] \end{aligned} \tag{4.25}$$

4.2.2 Maximum-Likelihood Estimation

The maximum-likelihood (ML) estimate $\hat{\boldsymbol{\theta}}_{ML}$ is obtained as the parameter vector that maximises the likelihood function $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$. The ML estimator corresponds to a Bayesian estimator with a uniform cost function and a uniform parameter prior pdf:

$$\begin{aligned}\mathcal{R}_{\mathrm{ML}}(\hat{\boldsymbol{\theta}}|\boldsymbol{y}) &= \int_{\boldsymbol{\theta}} [1-\delta(\hat{\boldsymbol{\theta}},\boldsymbol{\theta})] f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &= \text{const.}[1-f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\hat{\boldsymbol{\theta}})]\end{aligned} \tag{4.26}$$

where the prior function $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$=const. From a Bayesian point of view the main difference between the ML and MAP estimators is that the ML assumes that the prior pdf of $\boldsymbol{\theta}$ is uniform. Note that a uniform prior, in addition to modelling genuinely uniform pdfs, is also used when the parameter prior pdf is unknown, or when the parameter is an unknown constant.

From Equation (4.26), it is evident that minimisation of the risk function is achieved by maximisation of the likelihood function:

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \tag{4.27}$$

In practice it is convenient to maximise the log-likelihood function instead of the likelihood:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \log f_{\boldsymbol{Y}|\boldsymbol{\theta}}(\boldsymbol{Y}|\boldsymbol{\theta}) \tag{4.28}$$

The log-likelihood is usually chosen in practice because:

(a) the logarithm is a monotonic function, and hence the log-likelihood has the same turning points as the likelihood function;
(b) the joint log-likelihood of a set of independent variables is the sum of the log-likelihood of individual elements; and
(c) unlike the likelihood function, the log-likelihood has a dynamic range that does not cause computational under-flow.

Example 4.3 *ML Estimation of the mean and variance of a Gaussian process* Consider the problem of maximum likelihood estimation of the mean vector $\boldsymbol{\mu}_{\boldsymbol{y}}$ and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{yy}}$ of a P-dimensional

Gaussian vector process 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

$$f_Y(\boldsymbol{y}(0),\cdots,\boldsymbol{y}(N-1)) = \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2}\left|\boldsymbol{\Sigma}_{yy}\right|^{1/2}} \exp\left\{-\frac{1}{2}[\boldsymbol{y}(m)-\boldsymbol{\mu}_y]^{\mathrm{T}} \boldsymbol{\Sigma}_{yy}^{-1}[\boldsymbol{y}(m)-\boldsymbol{\mu}_y]\right\} \tag{4.29}$$

and the log-likelihood equation is given by

$$\ln f_Y(\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)) = \sum_{m=0}^{N-1}\left\{-\frac{P}{2}\ln(2\pi) - \frac{1}{2}\ln\left|\boldsymbol{\Sigma}_{yy}\right| - \frac{1}{2}[\boldsymbol{y}(m)-\boldsymbol{\mu}_y]^{\mathrm{T}} \boldsymbol{\Sigma}_{yy}^{-1}[\boldsymbol{y}(m)-\boldsymbol{\mu}_y]\right\} \tag{4.30}$$

Taking the derivative of the log-likelihood equation with respect to the mean vector $\boldsymbol{\mu}_y$ yields

$$\frac{\partial \ln f_Y(\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1))}{\partial \boldsymbol{\mu}_y} = \sum_{m=0}^{N-1}\left[2\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\mu}_y - 2\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{y}(m)\right] = 0 \tag{4.31}$$

From Equation (4.31), we have

$$\hat{\boldsymbol{\mu}}_y = \frac{1}{N}\sum_{m=0}^{N-1}\boldsymbol{y}(m) \tag{4.32}$$

To obtain the ML estimate of the covariance matrix we take the derivative of the log-likelihood equation with respect to $\boldsymbol{\Sigma}_{yy}^{-1}$:

$$\frac{\partial \ln f_Y(\boldsymbol{y}(0),\cdots,\boldsymbol{y}(N-1))}{\partial \boldsymbol{\Sigma}_{yy}^{-1}} = \sum_{m=0}^{N-1}\left\{\frac{1}{2}\boldsymbol{\Sigma}_{yy} - \frac{1}{2}[\boldsymbol{y}(m)-\boldsymbol{\mu}_y][\boldsymbol{y}(m)-\boldsymbol{\mu}_y]^{\mathrm{T}}\right\} = 0 \tag{4.33}$$

From Equation (4.31), we have an estimate of the covariance matrix as

$$\hat{\boldsymbol{\Sigma}}_{yy} = \frac{1}{N}\sum_{m=0}^{N-1}[\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_y][\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_y]^{\mathrm{T}} \tag{4.34}$$

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

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{\theta} + \boldsymbol{e} \tag{4.35}$$

where $\boldsymbol{e}$ is a random excitation input signal. The pdf of the parameter vector $\boldsymbol{\theta}$ given an observation vector $\boldsymbol{y}$ can be described, using Bayes' rule, as

$$f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}) f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \tag{4.36}$$

Assuming that the matrix $\boldsymbol{G}$ in Equation (4.35) is known, the likelihood of the signal $\boldsymbol{y}$ given the parameter vector $\boldsymbol{\theta}$ is the pdf of the random vector $\boldsymbol{e}$:

$$f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) = f_{\boldsymbol{E}}(\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta}) \tag{4.37}$$

Now assume the input $\boldsymbol{e}$ 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 $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}$. Therefore we have

$$f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}) = f_{\boldsymbol{E}}(\boldsymbol{e}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left[-\frac{1}{2\sigma_e^2} (\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{G}\boldsymbol{\theta}) \right] \tag{4.38}$$

and

$$f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{P/2} \left| \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} \right|^{1/2}} \exp\left[-\frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_{\boldsymbol{\theta}}) \right] \tag{4.39}$$

The ML estimate obtained from maximisation of the log-likelihood function $\ln\left[f_{Y|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}) \right]$ with respect to $\boldsymbol{\theta}$ is given by

$$\hat{\boldsymbol{\theta}}_{ML}(\boldsymbol{y}) = \left(\boldsymbol{G}^{\mathrm{T}} \boldsymbol{G} \right)^{-1} \boldsymbol{G}^{\mathrm{T}} \boldsymbol{y} \tag{4.40}$$

To obtain the MAP estimate we first form the posterior distribution by substituting Equations (4.38) and (4.39) in Equation (4.36)

$$f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \frac{1}{(2\pi\sigma_e^2)^{N/2}} \frac{1}{(2\pi)^{P/2}\left|\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}\right|^{1/2}} \times \exp\left(-\frac{1}{2\sigma_e^2}(\boldsymbol{y}-\boldsymbol{G\theta})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{G\theta}) - \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu}_{\boldsymbol{\theta}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu}_{\boldsymbol{\theta}}) \right) \tag{4.41}$$

The MAP parameter estimate is obtained by differentiating the log-likelihood function $\ln f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})$ and setting the derivative to zero:

$$\hat{\boldsymbol{\theta}}_{MAP}(\boldsymbol{y}) = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} + \sigma_e^2\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}\right)^{-1}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} + \sigma_e^2\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\theta}}\right) \tag{4.42}$$

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 (4.42) tends to the ML solution given by Equation (4.40). Conversely as the pdf of the parameter vector $\boldsymbol{\theta}$ becomes peaked, i.e. as $\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} \to 0$, the estimate tends towards $\boldsymbol{\mu}_{\boldsymbol{\theta}}$.

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 4.8) defined as

$$\begin{aligned} \mathcal{R}_{MMSE}(\hat{\boldsymbol{\theta}} \mid \boldsymbol{y}) &= \mathcal{E}[(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})^2 \mid \boldsymbol{y}] \\ &= \int_{\boldsymbol{\theta}} (\hat{\boldsymbol{\theta}}-\boldsymbol{\theta})^2 f_{\boldsymbol{\theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} \end{aligned} \tag{4.43}$$

In the following, it is shown that *the Bayesian MMSE estimate is the conditional mean of the posterior pdf*. Assuming that the mean square error risk function is differentiable and has a well-defined minimum, the MMSE solution can be obtained by setting the gradient of the mean square error risk function to zero:

$$\frac{\partial \mathcal{R}_{MMSE}(\hat{\boldsymbol{\theta}}|\boldsymbol{y})}{\partial \hat{\boldsymbol{\theta}}} = 2\hat{\boldsymbol{\theta}}\int_{\boldsymbol{\theta}} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} - 2\int_{\boldsymbol{\theta}} \boldsymbol{\theta}\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.44}$$

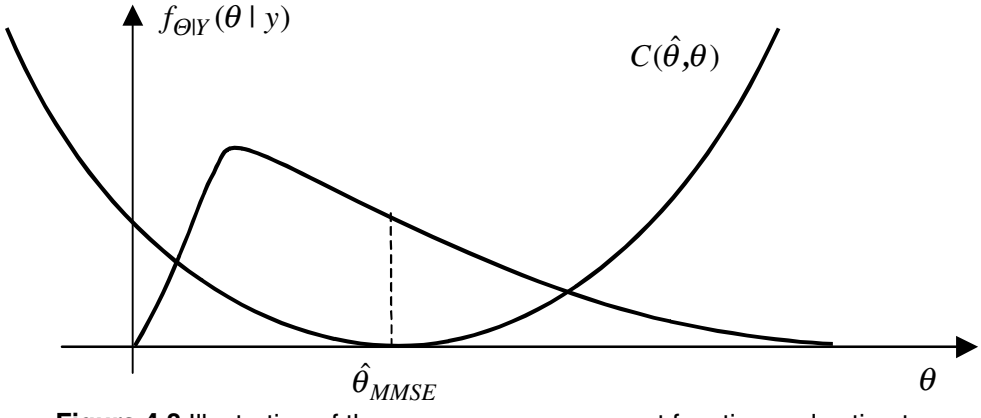


Figure 4.8 Illustration of the mean square error cost function and estimate.

Since the first integral on the right hand-side of Equation (4.42) is equal to 1, we have

$$\frac{\partial R_{MMSE}(\hat{\boldsymbol{\theta}} \mid \boldsymbol{y})}{\partial \hat{\boldsymbol{\theta}}} = 2\hat{\boldsymbol{\theta}} - \int_{\boldsymbol{\theta}} \boldsymbol{\theta}\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(d\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.45}$$

The MMSE solution is obtained by setting Equation (4.45) to zero:

$$\hat{\boldsymbol{\theta}}_{MMSE}(\boldsymbol{y}) = \int_{\boldsymbol{\theta}} \boldsymbol{\theta}\, f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y}) d\boldsymbol{\theta} \tag{4.46}$$

For cases where we do not have a pdf model of the parameter process, the minimum mean square error (known as the least square error, LSE) estimate is obtained through minimisation of a mean square error function $\mathcal{E}[e^2(\boldsymbol{\theta}|\boldsymbol{y})]$:

$$\hat{\boldsymbol{\theta}}_{LSE} = \arg\min_{\boldsymbol{\theta}} \mathcal{E}[e^2(\boldsymbol{\theta} \mid y)] \tag{4.47}$$

Th LSE estimation of Equation (4.47) does not use any prior knowledge of the distribution of the signals and the parameters. This can be considered as a strength of LSE in situations where the prior pdfs are unknown, but it can also be considered as a weakness in cases where fairly accurate models of the priors are available but not utilised.

Example 4.5 Consider the MMSE estimation of a parameter vector $\boldsymbol{\theta}$ assuming a linear model of the observation $\boldsymbol{y}$ as

$$\boldsymbol{y} = \boldsymbol{G}\boldsymbol{\theta} + \boldsymbol{e} \tag{4.48}$$

The LSE estimate is obtained as the parameter vector at which the gradient of the mean squared error with respect to $\boldsymbol{\theta}$ is zero:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} - 2\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}_{LSE}} = 0 \tag{4.49}$$

From Equation (4.49) the LSE parameter estimate is given by

$$\boldsymbol{\theta}_{LSE} = [\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}]^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} \tag{4.50}$$

Note that for a Gaussian likelihood function, the LSE solution is the same as the ML solution of Equation (4.40).

4.2.4 Minimum Mean Absolute Value of Error Estimation

The minimum mean absolute value of error (MAVE) estimate (Figure 4.9) is obtained through minimisation of a Bayesian risk function defined as

$$\mathcal{R}_{MAVE}(\hat{\boldsymbol{\theta}}\,|\boldsymbol{y}) = \mathcal{E}[|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\,|\,\boldsymbol{y}] = \int_{\boldsymbol{\theta}} |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\,| f_{\boldsymbol{\theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.51}$$

In the following it is shown that the minimum mean absolute value estimate is the median of the parameter process. Equation (4.51) can be re-expressed as

$$\mathcal{R}_{MAVE}(\hat{\boldsymbol{\theta}}\,|\boldsymbol{y}) = \int_{-\infty}^{\hat{\boldsymbol{\theta}}} [\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}] f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, d\boldsymbol{\theta} + \int_{\hat{\boldsymbol{\theta}}}^{\infty} [\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}] f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.52}$$

Taking the derivative of the risk function with respect to $\hat{\boldsymbol{\theta}}$ yields

$$\frac{\partial \mathcal{R}_{MAVE}(\hat{\boldsymbol{\theta}}\,|\boldsymbol{y})}{\partial \hat{\boldsymbol{\theta}}} = \int_{-\infty}^{\hat{\boldsymbol{\theta}}} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, d\boldsymbol{\theta} - \int_{\hat{\boldsymbol{\theta}}}^{\infty} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}\,|\,\boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.53}$$

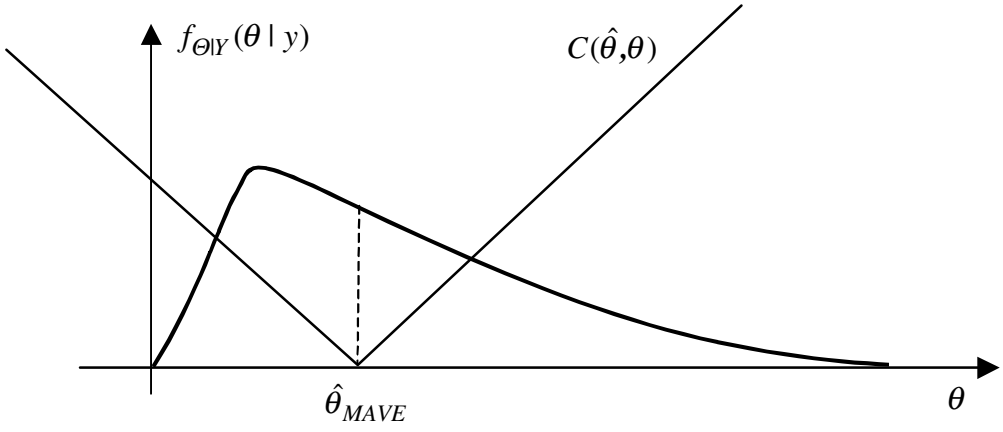


Figure 4.9 Illustration of mean absolute value of error cost function. Note that the MAVE estimate coincides with the conditional median of the posterior function.

The minimum absolute value of error is obtained by setting Equation (4.53) to zero:

$$\int_{-\infty}^{\hat{\boldsymbol{\theta}}_{MAVE}} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} = \int_{\hat{\boldsymbol{\theta}}_{MAVE}}^{\infty} f_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta} \mid \boldsymbol{y})\, d\boldsymbol{\theta} \tag{4.54}$$

From Equation (4.54) we note the MAVE estimate is the median of the posterior density.

4.2.5 Equivalence of the MAP, ML, MMSE and MAVE for Gaussian Processes With Uniform Distributed Parameters

Example 4.4 shows that for a Gaussian-distributed process the LSE estimate and the ML estimate are identical. Furthermore, Equation (4.42), for the MAP estimate of a Gaussian-distributed parameter, shows that as the parameter variance increases, or equivalently as the parameter prior pdf tends to a uniform distribution, the MAP estimate tends to the ML and LSE estimates. In general, for any symmetric distribution, centred round the maximum, the mode, the mean and the median are identical. Hence, for a process with a symmetric pdf, if the prior distribution of the parameter is uniform then the MAP, the ML, the MMSE and the MAVE parameter estimates are identical. Figure 4.10 illustrates a symmetric pdf, an asymmetric pdf, and the relative positions of various estimates.

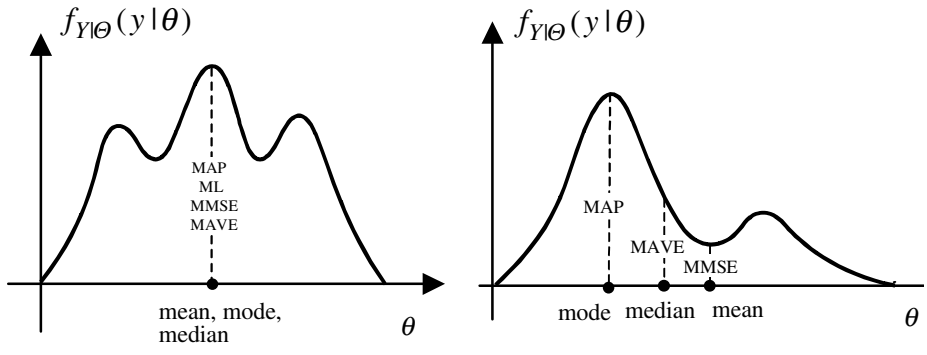


Figure 4.10 Illustration of a symmetric and an asymmetric pdf and their respective mode, mean and median and the relations to MAP, MAVE and MMSE estimates.

4.2.6 The Influence of the Prior on Estimation Bias and Variance

The use of a prior pdf introduces a bias in the estimate towards the range of parameter values with a relatively high 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 of a process are compared.

Example 4.6 Consider the ML estimation of a random scalar parameter θ, observed in a zero-mean additive white Gaussian noise (AWGN) $n(m)$, and expressed as

$$y(m) = \theta + n(m), \quad m = 0,..., N-1 \tag{4.55}$$

It is assumed that, for each realisation of the parameter θ, N observation samples are available. Note that, since the noise is assumed to be a zero-mean process, this problem is equivalent to estimation of the mean of the process $y(m)$. The likelihood of an observation vector $\boldsymbol{y}=[y(0), y(1), \ldots, y(N-1)]$ and a parameter value of θ is given by

$$\begin{aligned} f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y}|\theta) &= \prod_{m=0}^{N-1} f_N(y(m)-\theta) \\ &= \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2\right\} \end{aligned} \tag{4.56}$$

From Equation (4.56) the log-likelihood function is given by

$$\ln f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y}\,|\,\theta) = -\frac{N}{2}\ln(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2 \tag{4.57}$$

The ML estimate of θ, obtained by setting the derivative of $\ln f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y}|\theta)$ to zero, is given by

$$\hat{\theta}_{ML} = \frac{1}{N}\sum_{m=0}^{N-1} y(m) = \bar{y} \tag{4.58}$$

where $\bar{y}$ denotes the time average of $y(m)$. From Equation (4.56), we note that the ML solution is an unbiased estimate

$$\mathcal{E}[\hat{\theta}_{ML}] = \mathcal{E}\left(\frac{1}{N}\sum_{m=0}^{N-1}[\theta + n(m)]\right) = \theta \tag{4.59}$$

and the variance of the ML estimate is given by

$$\mathrm{Var}[\hat{\theta}_{ML}] = \mathcal{E}[(\hat{\theta}_{ML}-\theta)^2] = \mathcal{E}\left[\left(\frac{1}{N}\sum_{m=0}^{N-1} y(m) - \theta\right)^2\right] = \frac{\sigma_n^2}{N} \tag{4.60}$$

Note that the variance of the ML estimate decreases with increasing length of observation.

Example 4.7 *Estimation of a uniformly-distributed parameter observed in AWGN.* Consider the effects of using a uniform parameter prior on the mean and the variance of the estimate in Example 4.6. Assume that the prior for the parameter θ is given by

$$f_{\Theta}(\theta) = \begin{cases} 1/(\theta_{\max} - \theta_{\min}) & \theta_{\min} \le \theta \le \theta_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{4.61}$$

as illustrated in Figure 4.11. From Bayes' rule, the posterior pdf is given by

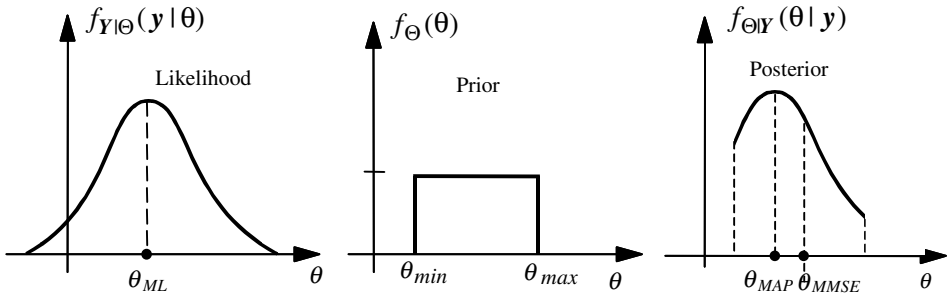


Figure 4.11 Illustration of the effects of a uniform prior.

$$f_{\Theta|\boldsymbol{Y}}(\theta \mid \boldsymbol{y}) = \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y} \mid \theta) f_{\Theta}(\theta)$$

$$= \begin{cases} \dfrac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \dfrac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left\{-\dfrac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2\right\}, & \theta_{\min} \le \theta \le \theta_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{4.62}$$

The MAP estimate is obtained by maximising the posterior pdf:

$$\hat{\theta}_{MAP}(\boldsymbol{y}) = \begin{cases} \theta_{\min} & \text{if } \hat{\theta}_{ML}(\boldsymbol{y}) < \theta_{\min} \\ \hat{\theta}_{ML}(\boldsymbol{y}) & \text{if } \theta_{\min} \ge \hat{\theta}_{ML}(\boldsymbol{y}) \ge \theta_{\max} \\ \theta_{\max} & \text{if } \hat{\theta}_{ML}(\boldsymbol{y}) > \theta_{\max} \end{cases} \tag{4.63}$$

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:

$$\mathrm{Var}[\hat{\theta}_{MAP}] = \int_{\theta_{\min}}^{\theta_{\max}} (\hat{\theta}_{MAP} - \theta)^2 f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y} \mid \theta \, d\boldsymbol{y} \le \mathrm{Var}[\hat{\theta}_{ML}] = \int_{-\infty}^{\infty} (\hat{\theta}_{ML} - \theta)^2 f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y} \mid \theta) \, d\boldsymbol{y} \tag{4.64}$$

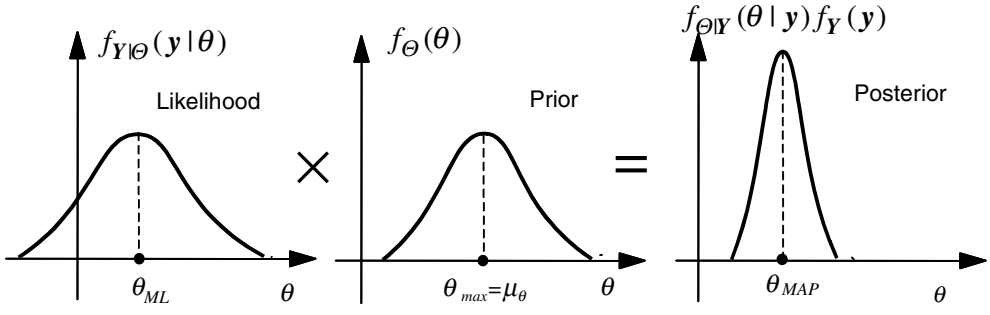


Figure 4.12 Illustration of the posterior pdf as product of the likelihood and the prior.

Example 4.8 *Estimation of a Gaussian-distributed parameter observed in AWGN.* In this example, we consider the effect of a Gaussian prior on the mean and the variance of the MAP estimate. Assume that the parameter θ is Gaussian-distributed with a mean μ_θ and a variance σ_θ^2 as

$$f_\Theta(\theta) = \frac{1}{(2\pi\sigma_\theta^2)^{1/2}} \exp\left[-\frac{(\theta-\mu_\theta)^2}{2\sigma_\theta^2}\right] \tag{4.65}$$

From Bayes rule the posterior pdf is given as the product of the likelihood and the prior pdfs as:

$$\begin{aligned} f_{\Theta|\boldsymbol{Y}}(\theta|\boldsymbol{y}) &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\Theta}(\boldsymbol{y}|\theta) f_\Theta(\theta) \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \frac{1}{(2\pi\sigma_n^2)^{N/2}(2\pi\sigma_\theta^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma_n^2}\sum_{m=0}^{N-1}[y(m)-\theta]^2 - \frac{1}{2\sigma_\theta^2}(\theta-\mu_\theta)^2\right\} \end{aligned} \tag{4.66}$$

The maximum posterior solution is obtained by setting the derivative of the log-posterior function, $\ln f_{\Theta|\boldsymbol{Y}}(\theta|\boldsymbol{y})$, with respect to θ to zero:

$$\hat{\theta}_{MAP}(\boldsymbol{y}) = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N}\,\bar{y} + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N}\,\mu_\theta \tag{4.67}$$

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 4.12. The expectation

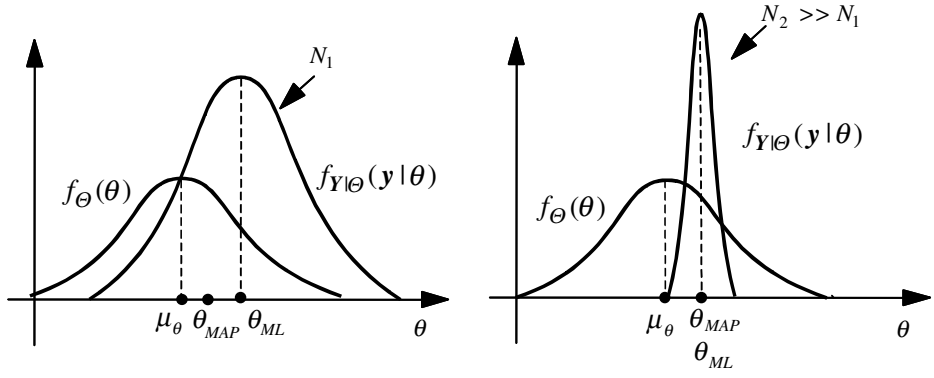


Figure 4.13 Illustration of the effect of increasing length of observation on the variance an estimator.

of the MAP estimate is obtained by noting that the only random variable on the right-hand side of Equation (4.67) is the term $\bar{y}$, and that $\mathcal{E}[\bar{y}]=\theta$

$$\mathcal{E}[\hat{\theta}_{MAP}(\mathbf{y})] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N}\theta + \frac{\sigma_n^2/N}{\sigma_\theta^2 + \sigma_n^2/N}\mu_\theta \tag{4.68}$$

and the variance of the MAP estimate is given as

$$\text{Var}[\hat{\theta}_{MAP}(\mathbf{y})] = \frac{\sigma_\theta^2}{\sigma_\theta^2 + \sigma_n^2/N} \times \text{Var}[\bar{y}] = \frac{\sigma_n^2/N}{1 + \sigma_n^2/N\sigma_\theta^2} \tag{4.69}$$

Substitution of Equation (4.58) in Equation (4.67) yields

$$\text{Var}[\hat{\theta}_{MAP}(\mathbf{y})] = \frac{\text{Var}[\hat{\theta}_{ML}(\mathbf{y})]}{1 + \text{Var}[\hat{\theta}_{ML}(\mathbf{y})]/\sigma_\theta^2} \tag{4.70}$$

Note that as σ_θ^2, the variance of the parameter θ, increases the influence of the prior decreases, and the variance of the MAP estimate tends towards the variance of the ML estimate.

4.2.7 The Relative Importance of the Prior and the Observation

A fundamental issue in the Bayesian inference method is the relative influence of the observation signal and the prior pdf on the outcome. The importance of the observation depends on the confidence in the observation, and the confidence in turn depends on the length of the observation and on

the signal-to-noise ratio (SNR). In general, as the number of observation samples and the SNR increase, the variance of the estimate and the influence of the prior decrease. From Equation (4.67) for the estimation of a Gaussian distributed parameter observed in AWGN, as the length of the observation N increases, the importance of the prior decreases, and the MAP estimate tends to the ML estimate:

$$\lim_{N\to\infty}\hat{\theta}_{MAP}(\boldsymbol{y})=\lim_{N\to\infty}\left(\frac{\sigma_\theta^2}{\sigma_\theta^2+\sigma_n^2/N}\bar{y}+\frac{\sigma_n^2/N}{\sigma_\theta^2+\sigma_n^2/N}\mu_\theta\right)=\bar{y}=\hat{\theta}_{ML} \tag{4.71}$$

As illustrated in Figure 4.13, as the length of the observation N tends to infinity then both the MAP and the ML estimates of the parameter should tend to its true value θ.

Example 4.9 *MAP estimation of a signal in additive noise.* Consider the estimation of a scalar-valued Gaussian signal $x(m)$, observed in an additive Gaussian white noise $n(m)$, and modelled as

$$y(m)=x(m)+n(m) \tag{4.72}$$

The posterior pdf of the signal $x(m)$ is given by

$$\begin{aligned} f_{X|Y}(x(m)|y(m)) &= \frac{1}{f_Y(y(m))}f_{Y|X}(y(m)|x(m))f_X(x(m)) \\ &= \frac{1}{f_Y(y(m))}f_N(y(m)-x(m))f_X(x(m)) \end{aligned} \tag{4.73}$$

where $f_X(x(m))=\mathcal{N}\left(x(m),\mu_x,\sigma_x^2\right)$ and $f_N(n(m))=\mathcal{N}\left(n(m),\mu_n,\sigma_n^2\right)$ are the Gaussian pdfs of the signal and noise respectively. Substitution of the signal and noise pdfs in Equation (4.73) yields

$$\begin{aligned} f_{X|Y}(x(m)\mid y(m)) = \frac{1}{f_Y(y(m))}\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left\{-\frac{[y(m)-x(m)-\mu_n]^2}{2\sigma_n^2}\right\} \\ \times\frac{1}{\sqrt{2\pi}\sigma_x}\exp\left\{-\frac{[x(m)-\mu_x]^2}{2\sigma_x^2}\right\} \end{aligned} \tag{4.74}$$

This equation can be rewritten as

$$f_{X|Y}(x(m)\mid y(m))=\frac{1}{f_Y(y(m))}\frac{1}{2\pi\sigma_n\sigma_x}\exp\left\{-\frac{\sigma_x^2[y(m)-x(m)-\mu_n]^2+\sigma_n^2[x(m)-\mu_x]^2}{2\sigma_x^2\sigma_n^2}\right\} \tag{4.75}$$

To obtain the MAP estimate we set the derivative of the log-likelihood function $\ln f_{X|Y}(x(m)\mid y(m))$ with respect to $x(m)$ to zero as

$$\frac{\partial[\ln f_{X|Y}(x(m)\mid y(m))]}{\partial\hat{x}(m)}=-\frac{-2\sigma_x^2(y(m)-x(m)-\mu_n)+2\sigma_n^2(x(m)-\mu_x)}{2\sigma_x^2\sigma_n^2}=0 \tag{4.76}$$

From Equation (4.76) the MAP signal estimate is given by

$$\hat{x}(m)=\frac{\sigma_x^2}{\sigma_x^2+\sigma_n^2}[y(m)-\mu_n]+\frac{\sigma_n^2}{\sigma_x^2+\sigma_n^2}\mu_x \tag{4.77}$$

Note that the estimate $\hat{x}(m)$ is a weighted linear interpolation between the unconditional mean of $x(m)$, μ_x, and the observed value $(y(m)–\mu_n)$. At a very poor SNR i.e. when $\sigma_x^2 << \sigma_n^2$ we have $\hat{x}(m)\approx\mu_x$; and, on the other hand, for a noise-free signal $\sigma_n^2=0$ and $\mu_n=0$ and we have $\hat{x}(m)=y(m)$.

Example 4.10 *MAP estimate of a Gaussian–AR process observed in AWGN.* Consider a vector of N samples $\boldsymbol{x}$ from an autoregressive (AR) process observed in an additive Gaussian noise, and modelled as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n} \tag{4.78}$$

From Chapter 8, a vector $\boldsymbol{x}$ from an AR process may be expressed as

$$\boldsymbol{e}=\boldsymbol{A}\boldsymbol{x} \tag{4.79}$$

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

$$f_X(\boldsymbol{x} \mid \boldsymbol{x}_0) = f_E(\boldsymbol{e}) = \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \tag{4.80}$$

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 $\boldsymbol{\Sigma}_{\boldsymbol{nn}}$, is given by

$$f_N(\boldsymbol{n}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_{\boldsymbol{nn}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{nn}}^{-1}\boldsymbol{n}\right) \tag{4.81}$$

From Bayes' rule, the pdf of the signal given the noisy observation is

$$f_{X|Y}(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{f_{Y|X}(\boldsymbol{y}|\boldsymbol{x}) f_X(\boldsymbol{x})}{f_Y(\boldsymbol{y})} = \frac{1}{f_Y(\boldsymbol{y})} f_N(\boldsymbol{y}-\boldsymbol{x}) f_X(\boldsymbol{x}) \tag{4.82}$$

Substitution of the pdfs of the signal and noise in Equation (4.82) yields

$$f_{X|Y}(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{1}{f_Y(\boldsymbol{y})(2\pi)^N \sigma_e^{N/2}|\boldsymbol{\Sigma}_{\boldsymbol{nn}}|^{1/2}} \exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{nn}}^{-1}(\boldsymbol{y}-\boldsymbol{x}) + \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\sigma_e^2}\right]\right\} \tag{4.83}$$

The MAP estimate corresponds to the minimum of the argument of the exponential function in Equation (4.83). Assuming that the argument of the exponential function is differentiable, and has a well-defined minimum, we can obtain the MAP estimate from

$$\hat{\boldsymbol{x}}_{MAP}(\boldsymbol{y}) = \arg\operatorname*{zero}_{\boldsymbol{x}}\left\{\frac{\partial}{\partial \boldsymbol{x}}\left[(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{nn}}^{-1}(\boldsymbol{y}-\boldsymbol{x}) + \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\sigma_e^2}\right]\right\} \tag{4.84}$$

The MAP estimate is

$$\hat{\boldsymbol{x}}_{MAP}(\boldsymbol{y}) = \left(\mathbf{I} + \frac{1}{\sigma_e^2}\boldsymbol{\Sigma}_{\boldsymbol{nn}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{y} \tag{4.85}$$

where $\mathbf{I}$ is the identity matrix.

4.3 The Estimate–Maximise (EM) Method

The EM algorithm is an iterative likelihood maximisation method with applications in blind deconvolution, model-based signal interpolation, spectral estimation from noisy observations, estimation of a set of model parameters from a training data set, etc. The EM is a framework for solving problems where it is difficult to obtain a direct ML estimate either because the data is incomplete or because the problem is difficult.

To define the term *incomplete data,* consider a signal $\boldsymbol{x}$ from a random process $\boldsymbol{X}$ with an unknown parameter vector $\boldsymbol{\theta}$ and a pdf $f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})$. The notation $f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})$ expresses the dependence of the pdf of $\boldsymbol{X}$ on the value of the unknown parameter $\boldsymbol{\theta}$. The signal $\boldsymbol{x}$ is the so-called *complete data* and the ML estimate of the parameter vector $\boldsymbol{\theta}$ may be obtained from $f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})$. 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) and is observed as $\boldsymbol{y}$. The observation $\boldsymbol{y}$ is the so-called incomplete data.
Maximisation of the likelihood of the incomplete data, $f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta})$, with respect to the parameter vector $\boldsymbol{\theta}$ is often a difficult task, whereas maximisation of the likelihood of the complete data $f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})$ 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 defined as

$$\mathcal{E}\left[\ln f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})\middle|\boldsymbol{y}\right] = \int_{\boldsymbol{X}} f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}) \ln f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})\, d\boldsymbol{x} \tag{4.86}$$

In Equation (4.86), the computation of the term $f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$ 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.

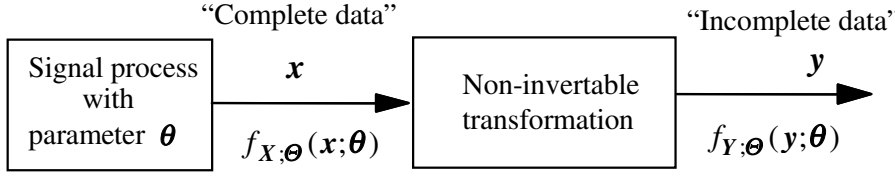


Figure 4.14 Illustration of transformation of complete data to incomplete data.

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

$$\begin{aligned} U(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_i) &= E[\ln f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta}) \mid \boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \\ &= \int_{\boldsymbol{X}} f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x} \mid \boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \ln f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})\, d\boldsymbol{x} \end{aligned} \tag{4.87}$$

Step 3: Maximisation Select

$$\hat{\boldsymbol{\theta}}_{i+1} = \arg\max_{\boldsymbol{\theta}} U(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_i) \tag{4.88}$$

Step 4: Convergence test If not converged then go to Step 2.

4.3.1 Convergence of the EM Algorithm

In this section, it is shown that the EM algorithm converges to a maximum of the likelihood of the incomplete data $f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta})$. The likelihood of the complete data can be written as

$$f_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}) = f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})\, f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta}) \tag{4.89}$$

where $f_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})$ is the likelihood of $\boldsymbol{x}$ and $\boldsymbol{y}$ with $\boldsymbol{\theta}$ as a parameter. From Equation (4.89), the log-likelihood of the incomplete data is obtained as

$$\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta}) = \ln f_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}) - \ln f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\,\boldsymbol{y};\boldsymbol{\theta}) \tag{4.90}$$

Using an estimate $\hat{\boldsymbol{\theta}}_i$ of the parameter vector $\boldsymbol{\theta}$, and taking the expectation of Equation (4.90) over the space of the complete signal $\boldsymbol{x}$, we obtain

$$\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta}) = U(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}_i) - V(\boldsymbol{\theta};\hat{\boldsymbol{\theta}}_i) \tag{4.91}$$

where for a given $\boldsymbol{y}$, the expectation of $\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta})$ is itself, and the function $U(\boldsymbol{\theta};\hat{\boldsymbol{\theta}})$ is the conditional expectation of $\ln f_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta})$:

$$\begin{aligned} U(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_i) &= \mathcal{E}[\ln f_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}) \mid \boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \\ &= \int_{\boldsymbol{X}} f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x} \mid \boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \ln f_{\boldsymbol{X};\boldsymbol{\Theta}}(\boldsymbol{x};\boldsymbol{\theta})\, d\boldsymbol{x} \end{aligned} \tag{4.92}$$

The function $V(\boldsymbol{\theta},\hat{\boldsymbol{\theta}})$ is the conditional expectation of $\ln f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})$:

$$\begin{aligned} V(\boldsymbol{\theta},\hat{\boldsymbol{\theta}}_i) &= \mathcal{E}\left[\ln f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{y};\hat{\boldsymbol{\theta}}_i\right] \\ &= \int_{\boldsymbol{X}} f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \ln f_{\boldsymbol{X}|\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\theta})\, d\boldsymbol{x} \end{aligned} \tag{4.93}$$

Now, from Equation (4.91), the log-likelihood of the incomplete data $\boldsymbol{y}$ with parameter estimate $\hat{\boldsymbol{\theta}}_i$ at iteration i is

$$\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\hat{\boldsymbol{\theta}}_i) = U(\hat{\boldsymbol{\theta}}_i;\hat{\boldsymbol{\theta}}_i) - V(\hat{\boldsymbol{\theta}}_i;\hat{\boldsymbol{\theta}}_i) \tag{4.94}$$

It can be shown (see Dempster et al., 1977) that the function V satisfies the inequality

$$V(\hat{\boldsymbol{\theta}}_{i+1};\hat{\boldsymbol{\theta}}_i) \leq V(\hat{\boldsymbol{\theta}}_i;\hat{\boldsymbol{\theta}}_i) \tag{4.95}$$

and in the maximisation step of EM we choose $\hat{\boldsymbol{\theta}}_{i+1}$ such that

$$U(\hat{\boldsymbol{\theta}}_{i+1};\hat{\boldsymbol{\theta}}_i) \geq U(\hat{\boldsymbol{\theta}}_i;\hat{\boldsymbol{\theta}}_i) \tag{4.96}$$

From Equation (4.94) and the inequalities (4.95) and (4.96), it follows that

$$\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\hat{\boldsymbol{\theta}}_{i+1}) \geq \ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\hat{\boldsymbol{\theta}}_i) \tag{4.97}$$

Therefore at every iteration of the EM algorithm, the conditional likelihood of the estimate increases until the estimate converges to a local maximum of the log-likelihood function $\ln f_{\boldsymbol{Y};\boldsymbol{\Theta}}(\boldsymbol{y};\boldsymbol{\theta})$.

The EM algorithm is applied to the solution of a number of problems in this book. In Section 4.5, of this chapter the estimation of the parameters of a mixture Gaussian model for the signal space of a recorded process is formulated in an EM framework. In Chapter 5, the EM is used for estimation of the parameters of a hidden Markov model.

4.4 Cramer–Rao Bound on the Minimum Estimator Variance

An important measure of the performance of an estimator is the variance of the estimate with the varying values of the observation signal $\boldsymbol{y}$ and the parameter vector $\boldsymbol{\theta}$. The minimum estimation variance depends on the distributions of the parameter vector $\boldsymbol{\theta}$ and on the observation signal $\boldsymbol{y}$. 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 estimate of the i^{th} coefficient θ_i of a parameter vector $\boldsymbol{\theta}$ is given as

$$\text{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq \frac{\left(1+\dfrac{\partial \theta_{\text{Bias}}}{\partial \theta_i}\right)^2}{\mathcal{E}\left[\left(\dfrac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right]} \tag{4.98}$$

An estimator that achieves the lower bound on the variance is called the minimum variance, or the most efficient, estimator.

Proof The bias in the estimate $\hat{\theta}_i(\boldsymbol{y})$ of the i^{th} coefficient of the parameter vector $\boldsymbol{\theta}$, averaged over the observation space $\boldsymbol{Y}$, is defined as

$$\mathcal{E}[\hat{\theta}_i(\boldsymbol{y}) - \theta_i] = \int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_i] f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} = \theta_{Bias} \tag{4.99}$$

Differentiation of Equation (4.99) with respect to θ_i yields

$$\int_{-\infty}^{\infty} \left\{ [\hat{\theta}_i(\boldsymbol{y}) - \theta_i] \frac{\partial f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta})}{\partial \theta_i} - f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta}) \right\} d\boldsymbol{y} = \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \tag{4.100}$$

For a probability density function we have

$$\int_{-\infty}^{\infty} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} = 1 \tag{4.101}$$

Therefore Equation (4.100) can be written as

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) \theta_i] \frac{\partial f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} d\boldsymbol{y} = 1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \tag{4.102}$$

Now, since the derivative of the integral of a pdf is zero, taking the derivative of Equation (4.101) and multiplying the result by θ_{Bias} yields

$$\theta_{\text{Bias}} \int_{-\infty}^{\infty} \frac{\partial f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} d\boldsymbol{y} = 0 \tag{4.103}$$

Substituting $\partial f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) / \partial \theta_i = f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) / \partial \theta_i$ into Equation (4.102), and using Equation (4.103), we obtain

$$\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_{\text{Bias}} - \theta_i] \frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y} = 1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \tag{4.104}$$

Now squaring both sides of Equation (4.104), we obtain

$$\left(\int_{-\infty}^{\infty} [\hat{\theta}_i(\boldsymbol{y}) - \theta_{\text{Bias}} - \theta_i] \frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y} \right)^2 = \left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \right)^2 \tag{4.105}$$

For the left-hand side of Equation (4.105) application of the following Schwartz inequality

$$\left(\int_{-\infty}^{\infty} f(y) g(y) dx \right)^2 \leq \int_{-\infty}^{\infty} (f(y))^2 dx \times \int_{-\infty}^{\infty} (g(y))^2 dy \tag{4.106}$$

yields

$$\left\{ \int_{-\infty}^{\infty} \left([\hat{\theta}_i(\boldsymbol{y}) - \theta_{\text{Bias}} - \theta_i] f_{\boldsymbol{Y}|\boldsymbol{\Theta}}^{1/2}(\boldsymbol{y}|\boldsymbol{\theta}) \right) \left(\frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}^{1/2}(\boldsymbol{y}|\boldsymbol{\theta}) \right) d\boldsymbol{y} \right\}^2 \leq$$

$$\left\{ \left(\int_{-\infty}^{\infty} \left([\hat{\theta}_i(\boldsymbol{y}) - \theta_{\text{Bias}} - \theta_i]^2 f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) \right) d\boldsymbol{y} \right) \right\} \left\{ \int_{-\infty}^{\infty} \left(\frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} \right)^2 f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y} \right\} \tag{4.107}$$

From Equations (4.105) and (4.107), we have

$$\text{Var}[\hat{\theta}_i(\boldsymbol{y})] \times \mathcal{E}\left[\left(\frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} \right)^2 \right] \geq \left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \right)^2 \tag{4.108}$$

The Cramer–Rao inequality (4.98) results directly from the inequality (4.108).

4.4.1 Cramer–Rao Bound for Random Parameters

For random parameters the Cramer–Rao bound may be obtained using the same procedure as above, with the difference that in Equation (4.98) instead of the likelihood $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$ we use the joint pdf $f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})$, and we also use the logarithmic relation

$$\frac{\partial \ln f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} + \frac{\partial \ln f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\partial \theta_i} \tag{4.109}$$

The Cramer–Rao bound for random parameters is obtained as

$$\text{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq \frac{\left(1 + \frac{\partial \theta_{\text{Bias}}}{\partial \theta_i} \right)^2}{\mathcal{E}\left[\left(\frac{\partial \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \theta_i} \right)^2 + \left(\frac{\partial \ln f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\partial \theta_i} \right)^2 \right]} \tag{4.110}$$

where the second term in the denominator of Equation (4.110) describes the effect of the prior pdf of $\boldsymbol{\theta}$. As expected the use of the prior, $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$, can result in a decrease in the variance of the estimate. An alternative form of the

minimum bound on estimation variance can be obtained by using the likelihood relation

$$\mathcal{E}\left[\left(\frac{\partial \ln f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i}\right)^2\right] = -\mathcal{E}\left[\frac{\partial^2 \ln f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i^2}\right] \tag{4.111}$$

as

$$\mathrm{Var}[\hat{\theta}_i(\boldsymbol{y})] \geq -\frac{\left(1+\dfrac{\partial \theta_{\mathrm{Bias}}}{\partial \theta_i}\right)^2}{\mathcal{E}\left[\dfrac{\partial^2 \ln f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}\,|\,\boldsymbol{\theta})}{\partial \theta_i^2} + \dfrac{\partial^2 \ln f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})}{\partial \theta_i^2}\right]} \tag{4.112}$$

4.4.2 Cramer–Rao Bound for a Vector Parameter

For real-valued P-dimensional vector parameters, the Cramer–Rao bound for the covariance matrix of an unbiased estimator of $\boldsymbol{\theta}$ is given by

$$\mathrm{Cov}[\hat{\boldsymbol{\theta}}] \geq \boldsymbol{J}^{-1}(\boldsymbol{\theta}) \tag{4.113}$$

where $\boldsymbol{J}$ is the $P \times P$ Fisher information matrix, with elements given by

$$[\boldsymbol{J}(\boldsymbol{\theta})]_{ij} = -\mathcal{E}\left[\frac{\partial^2 \ln f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\right] \tag{4.114}$$

The lower bound on the variance of the i^{th} element of the vector $\boldsymbol{\theta}$ is given by

$$\mathrm{Var}(\hat{\theta}_i) \geq \left[\boldsymbol{J}^{-1}(\boldsymbol{\theta})\right]_{ii} = \frac{1}{\mathcal{E}\left[\dfrac{\partial^2 \ln f_{\boldsymbol{Y},\boldsymbol{\Theta}}(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i^2}\right]} \tag{4.115}$$

where $(\boldsymbol{J}^{-1}(\boldsymbol{\theta})_{ii})$ is the i^{th} diagonal element of the inverse of the Fisher matrix.

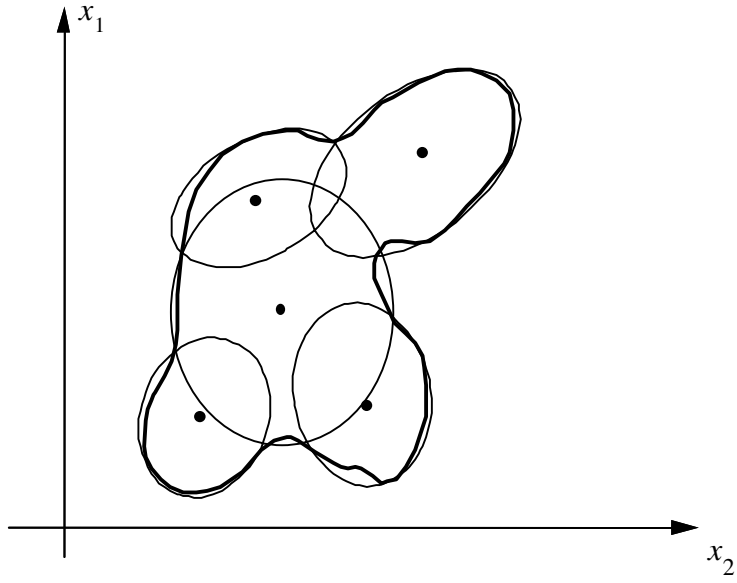


Figure 4.15 Illustration of probabilistic modelling of a two-dimensional signal space with a mixture of five bivariate Gaussian densities.

4.5 Design of Mixture Gaussian Models

A practical method for the modelling of 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. Figure 4.15 illustrates the 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 probabilities to different components of the Gaussian mixture.

A main advantage of the use of a mixture Gaussian model is that it results in mathematically tractable signal processing solutions. A mixture Gaussian pdf model for a process $\boldsymbol{X}$ is defined as

$$f_X(\boldsymbol{x}) = \sum_{k=1}^{K} P_k \mathcal{N}_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4.116}$$

where $\mathcal{N}_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the k^{th} component of the mixture Gaussian pdf, with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The parameter P_k is the

prior probability of the k^{th} mixture, and it can be interpreted as the expected fraction of the number of vectors from the process $\boldsymbol{X}$ associated with the k^{th} mixture.

In general, there are an infinite number of different K-mixture Gaussian densities that can be used to "tile up" a signal space. Hence the modelling of a signal space with a K-mixture pdf space can be regarded as a many-to-one mapping, and the expectation-maximisation (EM) method can be applied for the estimation of the parameters of the Gaussian pdf models.

4.5.1 The EM Algorithm for Estimation of Mixture Gaussian Densities

The EM algorithm, discussed in Section 4.4, 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 $[\boldsymbol{y}(m) \quad m=0, ..., N-1]$ 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 $\quad \boldsymbol{y}(m), \ m=0,\ldots,N-1$

The complete data $\quad \boldsymbol{x}(m)=[\boldsymbol{y}(m),k]=\boldsymbol{y}_k(m), \ m=0,\ldots,N-1, k\in(1,\ldots,K)$

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

$$\begin{aligned} U(\boldsymbol{\Theta},\hat{\boldsymbol{\Theta}}_i) &= \mathcal{E}[\ln f_{\boldsymbol{Y},K;\boldsymbol{\Theta}}(\boldsymbol{y}(m),k;\boldsymbol{\Theta}) \mid \boldsymbol{y}(m);\hat{\boldsymbol{\Theta}}_i] \\ &= \sum_{m=0}^{N-1}\sum_{k=0}^{K} \frac{f_{\boldsymbol{Y},K|\boldsymbol{\Theta}}(\boldsymbol{y}(m),k)\mid\hat{\boldsymbol{\Theta}}_i)}{f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}(m)\mid\hat{\boldsymbol{\Theta}}_i)} \ln f_{\boldsymbol{Y},K;\boldsymbol{\Theta}}(\boldsymbol{y}(m),k;\boldsymbol{\Theta}) \end{aligned} \tag{4.117}$$

where $\Theta=\{\boldsymbol{\theta}_k=[P_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k], k=1,..., K\}$, are the parameters of the Gaussian mixture as in Equation (4.116). Now the joint pdf of $\boldsymbol{y}(m)$ and the k^{th} Gaussian component of the mixture density can be written as

$$\begin{aligned} f_{\boldsymbol{Y},K|\boldsymbol{\Theta}}\left(\boldsymbol{y}(m), k\middle|\hat{\boldsymbol{\theta}}_i\right) &= P_{k_i} f_k\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\theta}}_{k_i}\right) \\ &= P_{k_i} \mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right) \end{aligned} \tag{4.118}$$

where $\mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)$ is a Gaussian density with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$:

$$\mathcal{N}_k\left(\boldsymbol{y}(m); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}(m)-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}(m)-\boldsymbol{\mu}_k)\right) \tag{4.119}$$

The pdf of $\boldsymbol{y}(m)$ as a mixture of K Gaussian densities is given by

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{\theta}}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\theta}}_i\right) &= \mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\theta}}_i\right) \\ &= \sum_{k=1}^{K} \hat{P}_{k_i} \mathcal{N}_k\left(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right) \end{aligned} \tag{4.120}$$

Substitution of the Gaussian densities of Equation (4.118) and Equation (4.120) in Equation (4.117) yields

$$U[(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{P}), (\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)] = \sum_{m=0}^{N-1} \sum_{k=1}^{K} \frac{\hat{P}_{k_i} \mathcal{N}_k(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)} \ln[P_k \mathcal{N}_k(\boldsymbol{y}(m); \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]$$

$$= \sum_{m=0}^{N-1} \sum_{k=1}^{K} \left(\frac{\hat{P}_{k_i} \mathcal{N}_k(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)} \ln P_k + \frac{\hat{P}_{k_i} \mathcal{N}_k(\boldsymbol{y}(m); \hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)} \ln \mathcal{N}_k(\boldsymbol{y}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right) \tag{4.121}$$

Equation (4.121) is maximised with respect to the parameter P_k using the constrained optimisation method. This involves subtracting the constant term $\Sigma P_k=1$ from the right hand side of Equation (4.121) and then setting the derivative of this equation with respect to P_k to zero, this yields

$$\begin{aligned}\hat{P}_{k_{i+1}} &= \arg\max_{P_k} U[(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{P}),(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)] \\ &= \frac{1}{N}\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)}\end{aligned} \tag{4.122}$$

The parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ that maximise the function U are obtained, by setting the derivative of the function with respect to these parameters to zero:

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{k_{i+1}} &= \arg\max_{\boldsymbol{\mu}_k} U[(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{P}),(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)] \\ &= \frac{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)}\boldsymbol{y}(m)}{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)}}\end{aligned} \tag{4.123}$$

and

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_{k_{i+1}} &= \arg\max_{\boldsymbol{\Sigma}_k} U[(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{P}),(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i, \hat{\boldsymbol{P}}_i)] \\ &= \frac{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)}(\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_{k_i})(\boldsymbol{y}(m)-\hat{\boldsymbol{\mu}}_{k_i})^T}{\sum_{m=0}^{N-1} \frac{\hat{P}_{k_i}\mathcal{N}_k(\boldsymbol{y}(m);\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i})}{\mathcal{N}\left(\boldsymbol{y}(m)\middle|\hat{\boldsymbol{\Theta}}_i\right)}}\end{aligned} \tag{4.124}$$

Equations (4.122)–(4.124) are the estimates of the parameters of a mixture Gaussian pdf model. These equations can be used in further iterations of the EM method until the parameter estimates converge.

4.6 Bayesian Classification

Classification is the processing and *labelling* of an observation sequence $\{\boldsymbol{y}(m)\}$ with one of M classes of signals $\{C_k;\ k=1,\ ...,\ M\}$ that could have generated the observation. Classifiers are present in all modern digital communication systems and in applications such as the decoding of

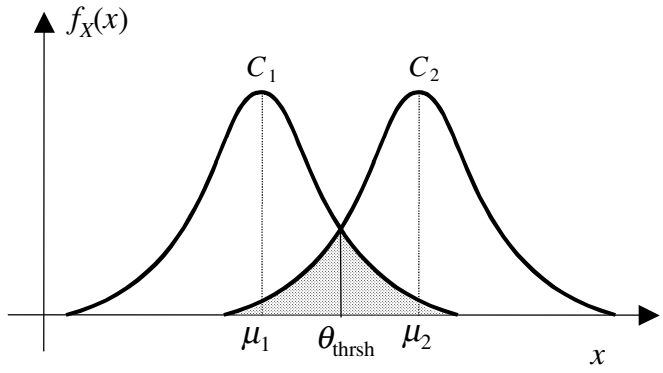


Figure 4.16 – Illustration of the overlap of the distribution of two classes of signals.

discrete-valued symbols in digital communication receivers, speech compression, video compression, speech recognition, image recognition, character recognition, signal/noise classification and detectors. For example, in an M-symbol digital communication system, the channel output signal is classified as one of the M signalling symbols; in speech recognition, segments of speech signals are labelled with one of about 40 elementary phonemes sounds; and in speech or video compression, a segment of speech samples or a block of image pixels 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, the bandwidth and the computational resources.

Classification errors are due to overlap of the distributions of different classes of signals. This is illustrated in Figure 4.16 for a binary classification problem with two Gaussian distributed signal classes C_1 and C_2. In the shaded region, where the signal distributions overlap, a sample x could belong to either of the two classes. The shaded area gives a measure of the classification error. The obvious solution suggested by Figure 4.16 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 adequately characterise each class and emphasise the difference 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 4.17(a) and 4.17(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 4.17(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 4.17(b) there is less distance between the mean of the sources, and the observation signals have a greater spread. This results 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:

$$P_{C|\boldsymbol{X}}(C_1|\boldsymbol{x}) \underset{C_2}{\overset{C_1}{\gtrless}} P_{C|\boldsymbol{X}}(C_2|\boldsymbol{x}) \tag{4.125}$$

Using Bayes' rule Equation (4.125) can be rewritten as

$$P_C(C_1) f_{\boldsymbol{X}|C}(\boldsymbol{x}|C_1) \underset{C_2}{\overset{C_1}{\gtrless}} P_C(C_2) f_{\boldsymbol{X}|C}(\boldsymbol{x}|C_2) \tag{4.126}$$

Letting $P_C(C_1)=P_1$ and $P_C(C_2)=P_2$, Equation (4.126) is often written in terms of a *likelihood ratio test* as

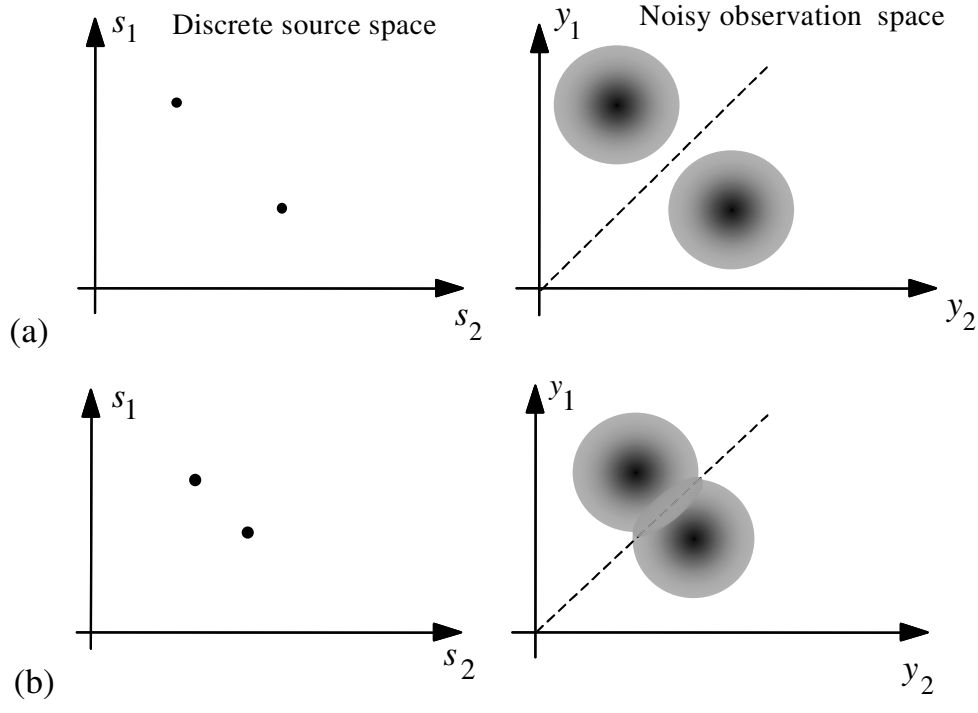


Figure 4.17 Illustration of binary classification: (a) the source and observation spaces are well separated, (b) the observation spaces overlap.

$$\frac{f_{\boldsymbol{X}|C}(\boldsymbol{x}|C_1)}{f_{\boldsymbol{X}|C}(\boldsymbol{x}|C_2)} \underset{C_2}{\overset{C_1}{\gtrless}} \frac{P_2}{P_1} \tag{4.127}$$

Taking the likelihood ratio yields the following discriminant function:

$$h(\boldsymbol{x}) = \ln f_{X|C}(\boldsymbol{x}|C_1) - \ln f_{X|C}(\boldsymbol{x}|C_2) \underset{C_2}{\overset{C_1}{\gtrless}} \ln \frac{P_2}{P_1} \tag{4.128}$$

Now assume that the signal in each class has a Gaussian distribution with a probability distribution function given by

$$f_{\boldsymbol{X}|C}(\boldsymbol{x}|c_i) = \frac{1}{\sqrt{2\pi}\,|\boldsymbol{\Sigma}_i|} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right], \quad \text{i=1,2} \tag{4.129}$$

From Equations (4.128) and (4.129), the discriminant function $h(\boldsymbol{x})$ becomes

$$h(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^{\mathrm{T}}\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} \underset{C_2}{\overset{C_1}{\gtrless}} \ln\frac{P_2}{P_1} \tag{4.130}$$

Example 4.10 For two Gaussian-distributed classes of scalar-valued signals with distributions given by $\mathcal{N}(x(m),\mu_1,\sigma^2)$ and $\mathcal{N}(x(m),\mu_2,\sigma^2)$, and equal class probability P_1=P_2=0.5, the discrimination function of Equation (4.130) becomes

$$h(x(m)) = \frac{\mu_2-\mu_1}{\sigma^2}x(m) + \frac{1}{2}\frac{\mu_2^2-\mu_1^2}{\sigma^2} \underset{C_2}{\overset{C_1}{\gtrless}} 0 \tag{4.131}$$

Hence the rule for signal classification becomes

$$x(m) \underset{C_2}{\overset{C_1}{\lessgtr}} \frac{\mu_1+\mu_2}{2} \tag{4.132}$$

The signal is labelled with class C_1 if $x(m)<(\mu_1+\mu_2)/2$ and as class C_2 otherwise.

4.6.2 Classification Error

Classification errors are due to the overlap of the distributions of different classes of signals. This is illustrated in Figure 4.16 for the binary classification of a scalar-valued signal and in Figure 4.17 for the binary classification of a two-dimensional signal. In each figure the overlapped area gives a measure of 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

$$P(Error|x) = P(C_1)P(x > Thrsh \mid x \in C_1) + P(C_2)P(x > Thrsh \mid x \in C_2) \quad (4.133)$$

For two Gaussian-distributed classes of scalar-valued signals with pdfs $\mathcal{N}(x(m),\mu_1,\sigma_1^2)$ and $\mathcal{N}(x(m),\mu_2,\sigma_2^2)$, Equation (4.133) becomes

$$P(Error|x) = P(C_1)\int_{Thrsh}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma_1}\exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)dx + P(C_2)\int_{-\infty}^{Thrsh} \frac{1}{\sqrt{2\pi}\sigma_2}\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)dx \quad (4.134)$$

where the parameter *Thrsh* is the classification threshold.

4.6.3 Bayesian Classification of Discrete-Valued Parameters

Let the set $\Theta=\{\boldsymbol{\theta}_i,\ i = 1,\ ...,\ M\}$ 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 Θ may be a discrete-valued or a continuous-valued space. Assuming that the observation space is continuous, the pdf of the parameter vector $\boldsymbol{\theta}_i$, given observation vector $\boldsymbol{y}$, may be expressed, using Bayes' rule, as

$$P_{\Theta|Y}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) = \frac{f_{Y|\Theta}(\boldsymbol{y}|\boldsymbol{\theta}_i)P_{\Theta}(\boldsymbol{\theta}_i)}{f_{Y}(\boldsymbol{y})} \quad (4.135)$$

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

$$\mathcal{R}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) = \sum_{j=1}^{M} C(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_j)P_{\Theta|Y}(\boldsymbol{\theta}_j \mid \boldsymbol{y}) \quad (4.136)$$

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 (4.136) can be

employed to obtain the maximum a posteriori, the maximum likelihood and the minimum mean square error classifiers.

4.6.4 Maximum A Posteriori Classification

MAP classification corresponds to Bayesian classification with a uniform cost function defined as

$$C(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_j) = 1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \tag{4.137}$$

where $\delta(\cdot)$ is the delta function. Substitution of this cost function in the Bayesian risk function yields

$$\begin{aligned} \mathcal{R}_{MAP}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) &= \sum_{j=1}^{M} [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)]\, P_{\boldsymbol{\Theta}|\boldsymbol{y}}(\boldsymbol{\theta}_j \mid \boldsymbol{y}) \\ &= 1 - P_{\boldsymbol{\Theta}|\boldsymbol{y}}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) \end{aligned} \tag{4.138}$$

Note that the MAP risk in selecting $\boldsymbol{\theta}_i$ is the classification error probability; that is the sum of the probabilities of all other candidates. From Equation (4.138) minimisation of the MAP risk function is achieved by maximisation of the posterior pmf:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{MAP}(\boldsymbol{y}) &= \arg\max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{\theta}_i} P_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_i) f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}_i) \end{aligned} \tag{4.139}$$

4.6.5 Maximum-Likelihood (ML) 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:

$$\begin{aligned} \mathcal{R}_{ML}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) &= \sum_{j=1}^{M} [1 - \delta(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)] \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}_j) P_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_j) \\ &= 1 - \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}_i) P_{\boldsymbol{\Theta}} \end{aligned} \tag{4.140}$$

where P_{Θ} is the uniform pmf of $\boldsymbol{\theta}$. Minimisation of the ML risk function (4.140) is equivalent to maximisation of the likelihood $f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}_i)$

$$\hat{\boldsymbol{\theta}}_{ML}(\boldsymbol{y}) = \arg\max_{\boldsymbol{\theta}_i} f_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y} \mid \boldsymbol{\theta}_i) \tag{4.141}$$

4.6.6 Minimum Mean Square Error Classification

The Bayesian minimum mean square error classification results from minimisation of the following risk function:

$$\mathcal{R}_{MMSE}(\boldsymbol{\theta}_i \mid \boldsymbol{y}) = \sum_{j=1}^{M} \left|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\right|^2 P_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}_j \mid \boldsymbol{y}) \tag{4.142}$$

For the case when $P_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}_j|\boldsymbol{y})$ is not available, the MMSE classifier is given by

$$\hat{\boldsymbol{\theta}}_{MMSE}(\boldsymbol{y}) = \arg\min_{\boldsymbol{\theta}_i} \left|\boldsymbol{\theta}_i - \boldsymbol{\theta}(\boldsymbol{y})\right|^2 \tag{4.143}$$

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 4.18 illustrates an M-state process, where the output of the i^{th} state is expressed as

$$\boldsymbol{x}(m) = h_i\left(\boldsymbol{\theta_i}, \boldsymbol{e}(m)\right), \quad i = 1, \ldots, M \tag{4.144}$$

where in each state the signal $\boldsymbol{x}(m)$ is modelled as the output of a state-dependent function $h_i(\cdot)$ with parameter $\boldsymbol{\theta}_i$, input $\boldsymbol{e}(m)$ and an input pdf $f_{Ei}(\boldsymbol{e}(m))$. The prior probability of each state is given by

$$P_S(s_i) = \mathcal{E}[N(s_i)] \Big/ \mathcal{E}\left[\sum_{j=1}^{M} N(s_j)\right] \tag{4.145}$$

where $\mathcal{E}[N(s_i)]$ is the expected number of 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

$$f_{\boldsymbol{X}}\left(\boldsymbol{x}(m)\right) = \sum_{i=1}^{M} P_S(s_i) f_{\boldsymbol{X}|S}(\boldsymbol{x} \mid s_i) \tag{4.146}$$

In Figure 4.18, 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

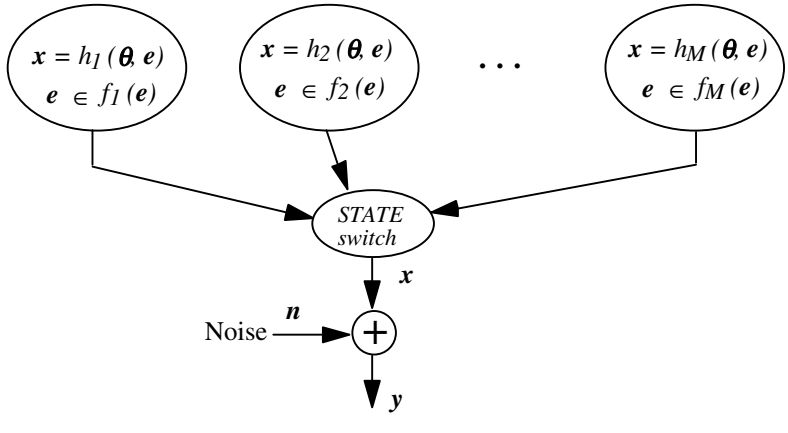


Figure 4.18 Illustration of a random process generated by a finite state system.

$$P_{S|\boldsymbol{Y}}(s_i|\boldsymbol{y}(m)) = \frac{f_{\boldsymbol{Y}|S}(\boldsymbol{y}(m)|s_i)P_S(s_i)}{\sum_{j=1}^{M} f_{\boldsymbol{Y}|S}(\boldsymbol{y}(m)|s_j)P_S(s_j)} \tag{4.147}$$

In MAP classification, the state with the maximum posterior probability is selected as

$$s_{MAP}(\boldsymbol{y}(m)) = \arg\max_{s_i} P_{S|\boldsymbol{Y}}(s_i|\boldsymbol{y}(m)) \tag{4.148}$$

The Bayesian state classifier assigns a misclassification cost function $C(s_i|s_j)$ to the action of selecting the state s_i when the true state is s_j. The risk function for the Bayesian classification is given by

$$\mathcal{R}(s_i|\boldsymbol{y}(\mathrm{m})) = \sum_{j=1}^{M} C(s_i|s_j)P_{S|\boldsymbol{Y}}(s_j|\boldsymbol{y}(\mathrm{m})) \tag{4.149}$$

4.6.8 Bayesian Estimation of the Most Likely State Sequence

Consider the estimation of the most likely state sequence $\boldsymbol{s} = [s_{i_0}, s_{i_1}, \ldots, s_{i_{T-1}}]$ of a finite state process, given a sequence of T observation vectors $\boldsymbol{Y} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}]$. A state sequence $\boldsymbol{s}$, of length T, is itself a random integer-valued vector process with N^T possible values. From the Bayes rule, the posterior pmf of a state sequence $\boldsymbol{s}$, given an observation sequence $\boldsymbol{Y}$, can be expressed as

$$P_{S|Y}(s_{i_0}, \ldots, s_{i_{T-1}} | \boldsymbol{y}_0, \ldots, \boldsymbol{y}_{T-1}) = \frac{f_{Y|S}(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{T-1} | s_{i_0}, \ldots, s_{i_{T-1}})P_S(s_{i_0}, \ldots, s_{i_{T-1}})}{f_Y(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{T-1})} \tag{4.150}$$

where $P_{\boldsymbol{S}}(\boldsymbol{s})$ is the pmf of the state sequence $\boldsymbol{s}$, and for a given observation sequence, the denominator $f_Y(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{T-1})$ is a constant. The Bayesian risk in selecting a state sequence $\boldsymbol{s}_i$ is expressed as

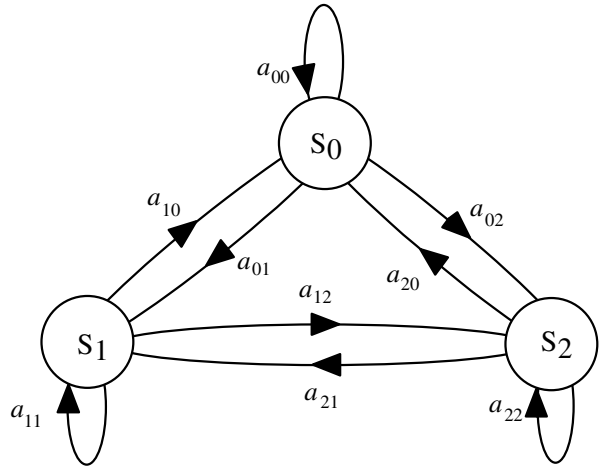


Figure 4.19 A three state Markov Process.

$$\mathcal{R}(\boldsymbol{s}_i|\boldsymbol{y}) = \sum_{j=1}^{N^T} C(\boldsymbol{s}_i \mid \boldsymbol{s}_j) P_{S|\boldsymbol{Y}}(\boldsymbol{s}_j|\boldsymbol{y}) \tag{4.151}$$

For a statistically independent process, the state of the process at any time is independent of the previous states, and hence the conditional probability of a state sequence can be written as

$$P_{S|Y}(s_{i_0},\ldots,s_{i_{T-1}}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{T-1}) = \prod_{k=0}^{T-1} f_{Y|S}(\boldsymbol{y}_k|s_{i_k}) P_S(s_{i_k}) \tag{4.152}$$

where s_{ik} denotes state s_i at time instant k. A particular case of a finite state process is the Markov chain 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

$$P_{S|Y}(s_{i_0},\ldots,s_{i_{T-1}} \mid \boldsymbol{y}_0,\ldots,\boldsymbol{y}_{T-1}) = \prod_{k=0}^{T-1} a_{i_{k-1}i_k} f_{S|Y}(s_{i_k} \mid \boldsymbol{y}_k) \tag{4.153}$$

where $a_{i_{k-1}i_k}$ is the probability that the process moves from state $s_{i_{k-1}}$ to state s_{i_k} Finite state random processes and computationally efficient methods of state sequence estimation are described in detail in Chapter 5.

4.7 Modelling the Space of a Random Process

In this section, we consider the training of statistical models for a database of P-dimensional vectors of a random process. The vectors in the database can be visualised as forming a number of clusters or regions 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

In vector quantisation, the space of a random vector process $\boldsymbol{X}$ is partitioned into K clusters or regions $[\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K]$, and each cluster $\boldsymbol{X}_i$ is represented by a cluster centroid $\boldsymbol{c}_i$. The set of centroid vectors $[\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_K]$ form a VQ code book model of the process $\boldsymbol{X}$. The VQ code book 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

$$Label(\boldsymbol{x}) = \arg\min_i d(\boldsymbol{x}, \boldsymbol{c}_i) \tag{4.154}$$

where $d(\boldsymbol{x}, \boldsymbol{c}_i)$ is a measure of distance between the vectors $\boldsymbol{x}$ and $\boldsymbol{c}_i$. The most commonly used distance measure is the mean squared distance.

4.7.2 Design of a Vector Quantiser: *K*-Means Clustering

The K-means algorithm, illustrated in Figure 4.20, is an iterative method for the design of a VQ codebook. Each iteration consists of two basic steps : (a) Partition the training signal space into K regions or clusters and (b) compute the centroid of each region. The steps in K-Means method are as follows:

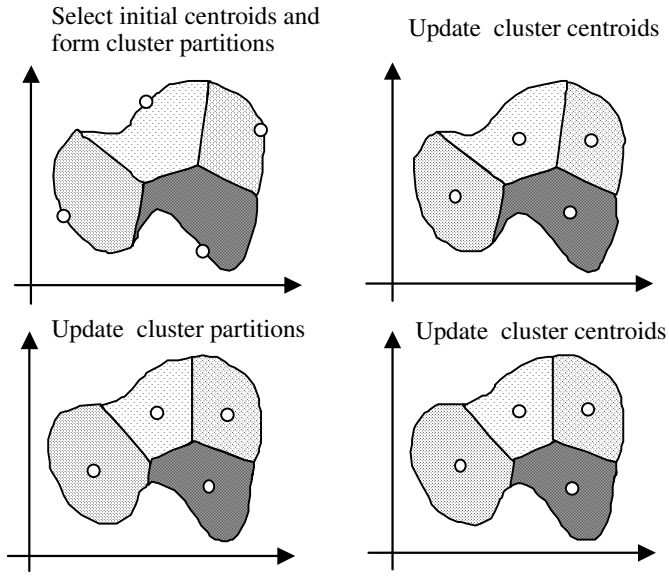


Figure 4.18 Illustration of the *K*-means clustering method.

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

$$D_i(m) = \frac{1}{N_i} \sum_{j=1}^{N_i} d(\boldsymbol{x}_i(j), \boldsymbol{c}_i(m)) \tag{4.155}$$

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

$$D(m) = \sum_{i=1}^{K} D_i(m) \tag{4.156}$$

Step 4: *Convergence test*:

if

$$D(m-1) - D(m) \geq Threshold \quad \text{stop,}$$

else

goto Step 2.

A vector quantiser models the regions, or the clusters, of the signal space with a set of cluster centroids. A more complete description of the signal space can be achieved by modelling each cluster with a Gaussian density as described in the next chapter.

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 blind equalisation of distorted signals studied in Chapter 15.

We considered a number of examples of the estimation of a signal observed in noise, and derived the expressions for the effects of using prior pdfs on the mean and the variance of the estimates. The choice of the prior pdf is an important consideration in Bayesian estimation. Many processes, for example speech or the response of a 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 that is the subspace consistent with both the likelihood and the prior pdfs. The choice of the prior, depending on how well it fits the process, can have a significant influence on the solutions.

The iterative 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 modelling of a data space with a mixture Gaussian process, and used the EM method to derive a solution for the parameters of the mixture Gaussian model.

Bibliography

ANDERGERG M.R. (1973) Cluster Analysis for Applications. Academic Press, New York.

ABRAMSON N. (1963) Information Theory and Coding. McGraw Hill, 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**, pp.164–171.

BAYES T. (1763) An Essay Towards Solving a Problem in the Doctrine of Changes, Phil. Trans. Royal Society of London, **53**, pp. 370–418, (reprinted in 1958 in Biometrika, **45**, pp. 293–315).

CHOU P. LOOKABAUGH T. and GRAY R. (1989) Entropy-Constrained Vector Quantisation. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-37**, pp. 31–42.

BEZDEK J.C. (1981) Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.

CRAMER H. (1974) Mathematical Methods of Statistics. Princeton University Press.

DEUTSCH R. (1965) Estimation Theory. Prentice-Hall, Englewood Cliffs, NJ.

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**, pp. 1-38.

DUDA R.O. and HART R.E. (1973) Pattern Classification. Wiley, New York.

FEDER M. and WEINSTEIN E. (1988) Parameter Estimation of Superimposed Signals using the EM algorithm. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-36(4)**, pp. 477.489.

FISHER R.A. (1922) On the Mathematical Foundations of the Theoretical Statistics. Phil Trans. Royal. Soc. London, **222**, pp. 309–368.

GERSHO A. (1982) On the Structure of Vector Quantisers. IEEE Trans. Information Theory, **IT-28**, pp. 157–166.

GRAY R.M. (1984) Vector Quantisation. IEEE ASSP Magazine, p. 4-29.

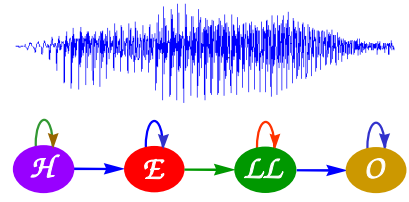
GRAY R.M. and KARNIN E.D (1982), Multiple local Optima in Vector Quantisers. IEEE Trans. Information Theory, **IT-28**, pp. 256–261.

JEFFREY H. (1961) Scientific Inference, 3rd ed. Cambridge University Press.
LARSON H.J. and BRUNO O.S. (1979) Probabilistic Models in Engineering Sciences. **I** and **II.** Wiley, New York.
LINDE Y., BUZO A. and GRAY R.M. (1980) An Algorithm for Vector Quantiser Design. IEEE Trans. Comm. **COM-28**, pp. 84–95.
MAKHOUL J., ROUCOS S., and GISH H. (1985) Vector Quantisation in Speech Coding. Proc. IEEE, **73**, pp. 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**, pp. 81–91.
RENDER R.A. and WALKER H.F.(1984) Mixture Densities, Maximum Likelihood and the EM algorithm. SIAM review, **26**, pp. 195–239.
SCHARF L.L. (1991) Statistical Signal Processing: Detection, Estimation, and Time Series Analysis. Addison Wesley, Reading, MA.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

5



HIDDEN MARKOV MODELS

Hidden Markov models (HMMs) are used for the statistical modelling of non-stationary signal processes such as speech signals, image sequences and time-varying noise. An HMM models the time variations (and/or the space variations) of the statistics of a random process with a Markovian chain of state-dependent stationary subprocesses. An HMM is essentially a Bayesian finite state process, with a Markovian prior for modelling the transitions between the states, and a set of state probability density functions for modelling the random variations of the signal process within each state. This chapter begins with a brief introduction to continuous and finite state 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 the HMMs for the enhancement of noisy signals are considered.

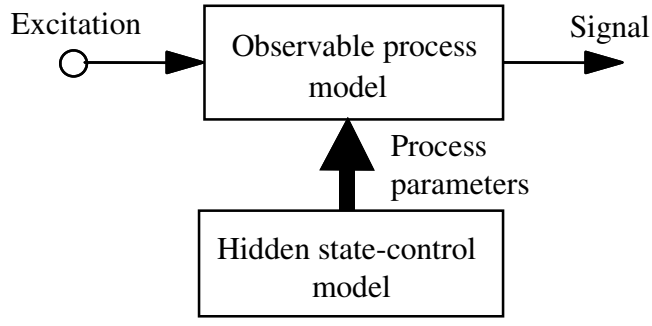


Figure 5.1 Illustration of a two-layered model of a non-stationary process.

5.1 Statistical Models for Non-Stationary Processes

A non-stationary process can be defined as one whose statistical parameters vary over time. Most "naturally generated" signals, such as audio signals, image 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.

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:

(a) *Continuously variable state* processes.
(b) *Finite state* processes.

A continuously variable state process is defined as one whose underlying statistics vary continuously with time. Examples of this class of random processes are audio signals such as speech and music, whose power and spectral composition 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. Continuously variable processes can be approximated by an appropriate finite state 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

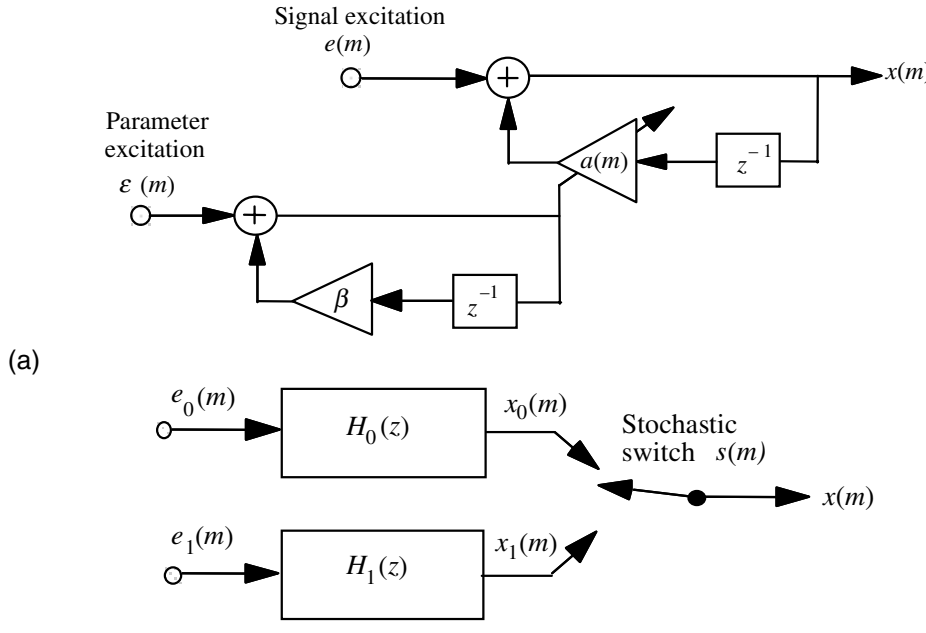


Figure 5.2 (a) A continuously variable state AR process. (b) A binary-state AR process.

parameters of the non-stationary AR model. For this model, the observation signal equation and the parameter state equation can be expressed as

$$x(m) = a(m)x(m-1) + e(m) \qquad \text{Observation equation} \qquad (5.1)$$

$$a(m) = \beta a(m-1) + \varepsilon(m) \qquad \text{Hidden state equation} \qquad (5.2)$$

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 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

$$x(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \qquad (5.3)$$

where the binary switch $s(m)$ selects the state of the process at time m, and $\bar{s}(m)$ denotes the Boolean complement of $s(m)$.

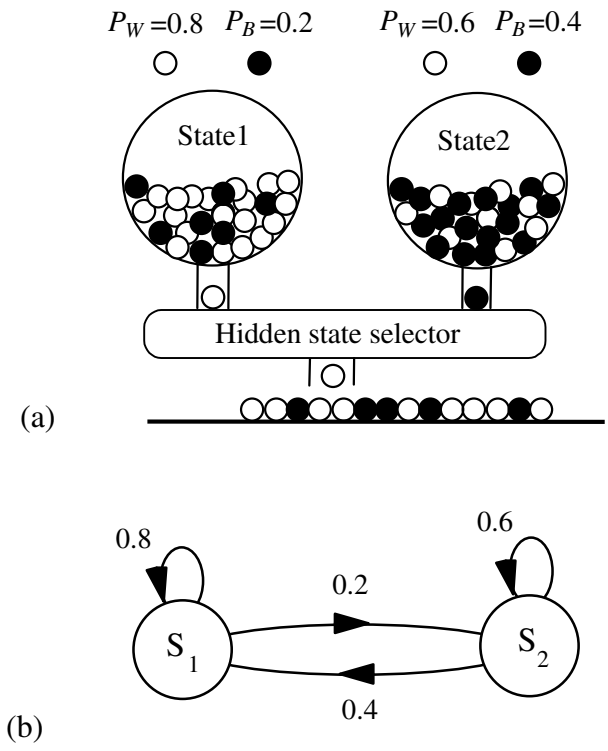


Figure 5.3 (a) Illustration of a two-layered random process. (b) An HMM model of the process in (a).

5.2 Hidden Markov Models

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, 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 above Figure 5.3. 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 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(b). Note that in this example the observable outcome does not unambiguously indicate the underlying hidden state, because both states are capable of releasing black and white balls.

In general, a hidden Markov model has N sates, 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. A general form of a three-state HMM is shown in Figure 5.4. This structure is known as an *ergodic* HMM. In the context of an HMM, the term "ergodic" implies that there are no structural constraints for connecting any state to any other state.

A more constrained form of an HMM is the left–right model of Figure 5.5, so-called because the allowed state transitions are those from a left state to a right state and the self-loop transitions. The left–right constraint is useful for the characterisation of temporal or sequential structures of stochastic signals such as speech and musical signals, because time may be visualised as having a direction from left to right.

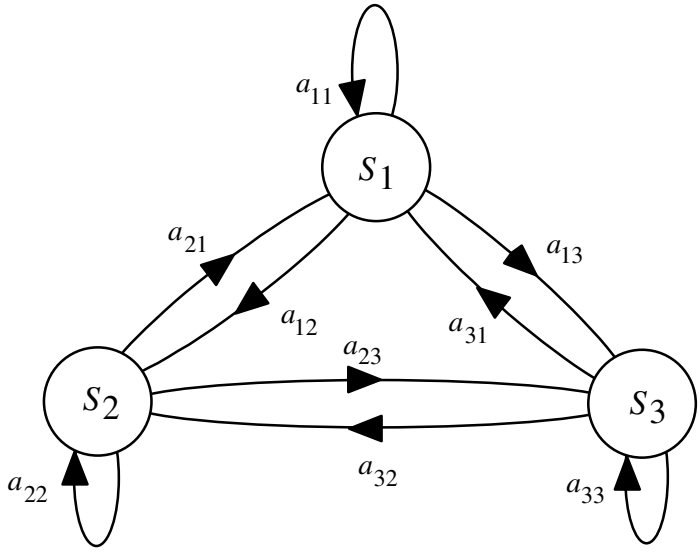


Figure 5.4 A three-state ergodic HMM structure.

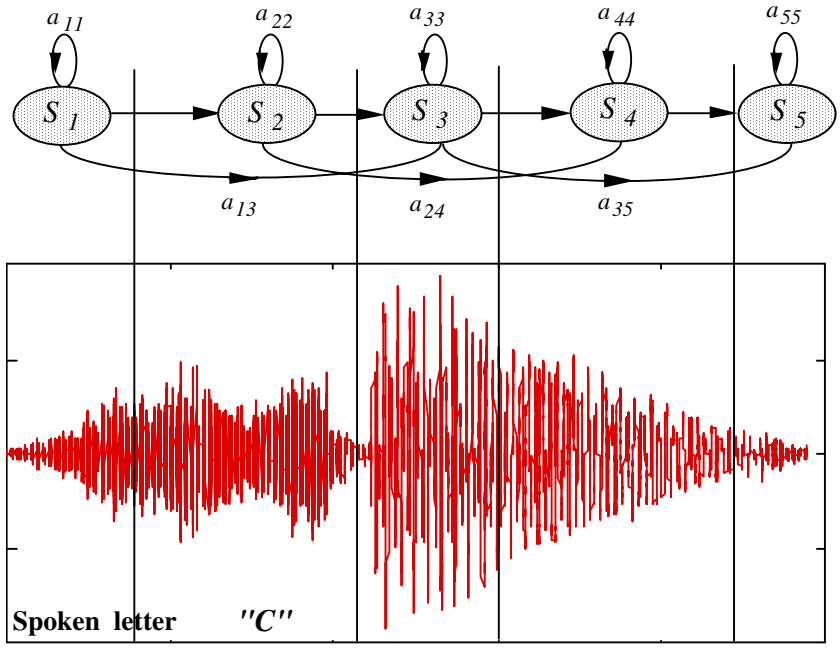


Figure 5.5 A 5-state left–right HMM speech model.

5.2.1 A Physical Interpretation of Hidden Markov Models

For a physical interpretation of the use of HMMs in modelling a signal process, consider the illustration of Figure 5.5 which shows a left–right HMM of a spoken letter "C", phonetically transcribed 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. 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 the 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 probability distributions of the spectral composition of the signal segments associated with each state.

5.2.2 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* pmf of a state sequence $\boldsymbol{s}$ of a model $\mathcal{M}$, given an observation sequence $\boldsymbol{X}$, can be expressed using Bayes' rule as the product of a state *prior* pmf and an observation *likelihood* function:

$$P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(\boldsymbol{s}|\boldsymbol{X},\mathcal{M}) = \frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})} P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}) f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M}) \tag{5.4}$$

where the observation sequence $\boldsymbol{X}$ is modelled by a probability density function $P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(\boldsymbol{s}|\boldsymbol{X},\mathcal{M})$.

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 $P_{\mathcal{M}}(\mathcal{M})$:

$$P_{\mathcal{M}|\boldsymbol{X}}(\mathcal{M}|\boldsymbol{X}) = \frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})} \underbrace{P_{\mathcal{M}}(\mathcal{M})}_{\textit{Model prior}} \sum_{\boldsymbol{s}} \underbrace{P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})}_{\textit{State prior}} \underbrace{f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})}_{\textit{Observation likelihood}} \tag{5.5}$$

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.

5.2.3 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 isolated-word speech modelling N is set between 5 to 10.

State transition-probability matrix $\boldsymbol{A}=\{a_{ij},\ i,j=1,\ ...\ N\}$. 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 $\{\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, ..., \boldsymbol{\mu}_{iM}, i=1, ..., N\}$. For each state a set of M prototype vectors model the centroids of the signal space associated with each state.

State observation vector probability model. This can be either a discrete model composed of the M prototype vectors and their associated probability mass function (pmf) $\boldsymbol{P}=\{P_{ij}(\cdot);\ i=1, ..., N, j=1, ...\ M\}$, or it may be a continuous (usually Gaussian) pdf model $\boldsymbol{F}=\{f_{ij}(\cdot);\ i=1, ..., N, j=1, ..., M\}$.

Initial state probability vector $\boldsymbol{\pi}=[\pi_1, \pi_2, ..., \pi_N]$.

5.2.4 State Observation Models

Depending on whether a signal process is discrete-valued or continuous-valued, the state observation model for the process can be either a discrete-valued probability mass function (pmf), or a continuous-valued probability density function (pdf). The discrete models can also be used for the modelling of the space of a continuous-valued process quantised into a number of discrete points. First, consider a discrete state observation density model. Assume that associated with the i^{th} state of an HMM there are M discrete centroid vectors $[\boldsymbol{\mu}_{i1}, ..., \boldsymbol{\mu}_{iM}]$ with a pmf $[P_{i1}, ..., P_{iM}]$. These centroid vectors and their probabilities are normally obtained through clustering of a set of training signals associated with each state.

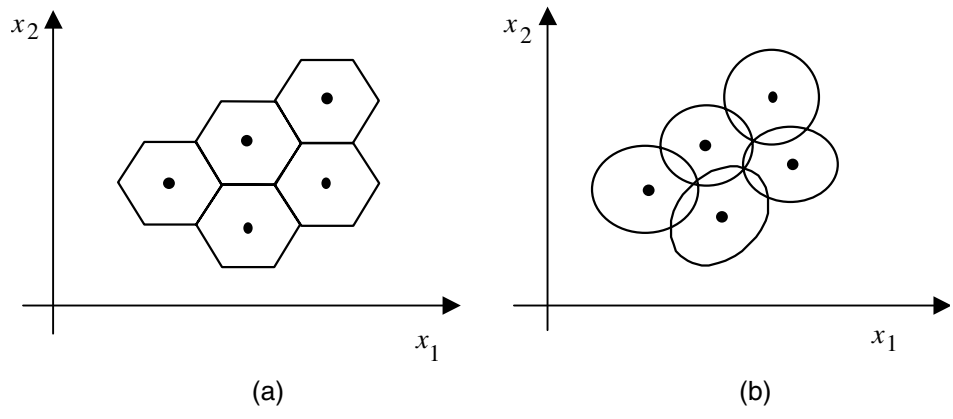


Figure 5.6 Modelling a random signal space using (a) a discrete-valued pmf and (b) a continuous-valued mixture Gaussian density.

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 $\{\boldsymbol{\mu}_{ik}, P_{ik}; i=1, ..., N, k=1, ..., M\}$. In effect, this results in a discrete state observation HMM for a continuous-valued process. Figure 5.6(a) shows a partitioning, and quantisation, of a signal space into a number of centroids.

Now if each cluster of the state observation space is modelled by a continuous pdf, such as a Gaussian pdf, then a continuous density HMM results. The most widely used state observation pdf for an HMM is the mixture Gaussian density defined as

$$f_{\boldsymbol{X}|S}(\boldsymbol{x}|s=i)=\sum_{k=1}^{M} P_{ik}\,\mathcal{N}(\boldsymbol{x},\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik}) \tag{5.6}$$

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 mixture pdf for the state i. Figure 5.6(b) shows the space of a mixture Gaussian model of an observation signal space. A 5-mode mixture Gaussian pdf is shown in Figure 5.7.

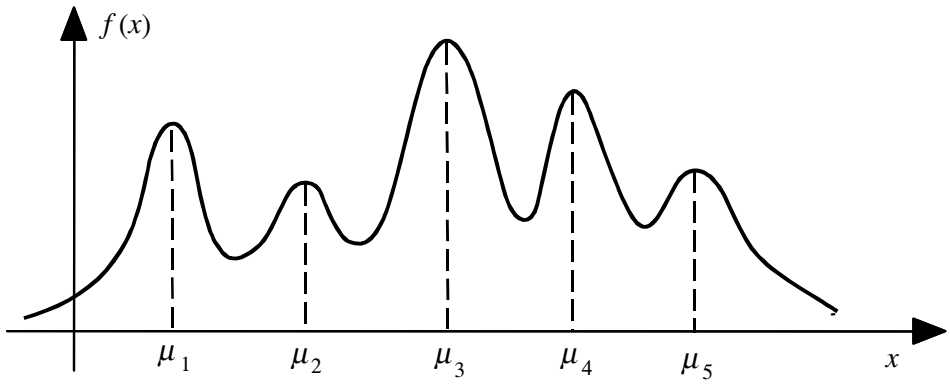


Figure 5.7 A mixture Gaussian probability density function.

5.2.5 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

$$\begin{aligned} &Prob\big(s(t)=j \,\big|\, s(t-1)=i, s(t-2)=k,\ldots, s(t-N)=l\big) \\ &\quad = Prob\big(s(t)=j \,\big|\, s(t-1)=i\big) = a_{ij} \end{aligned} \tag{5.7}$$

where $s(t)$ denotes the state of HMM at time t. The transition probabilities provide a probabilistic mechanism for connecting the states of an HMM, and for modelling the variations in the duration of the signals associated with each state. The probability of occupancy of a state i for d consecutive time units, $P_i(d)$, can be expressed in terms of the state self-loop transition probabilities a_{ii} as

$$P_i(d) = a_{ii}^{d-1}(1-a_{ii}) \tag{5.8}$$

From Equation (5.8), using the geometric series conversion formula, the mean occupancy duration for each state of an HMM can be derived as

$$\text{Mean occupancy of state } i = \sum_{d=0}^{\infty} d\,P_i(d) = \frac{1}{1-a_{ii}} \tag{5.9}$$

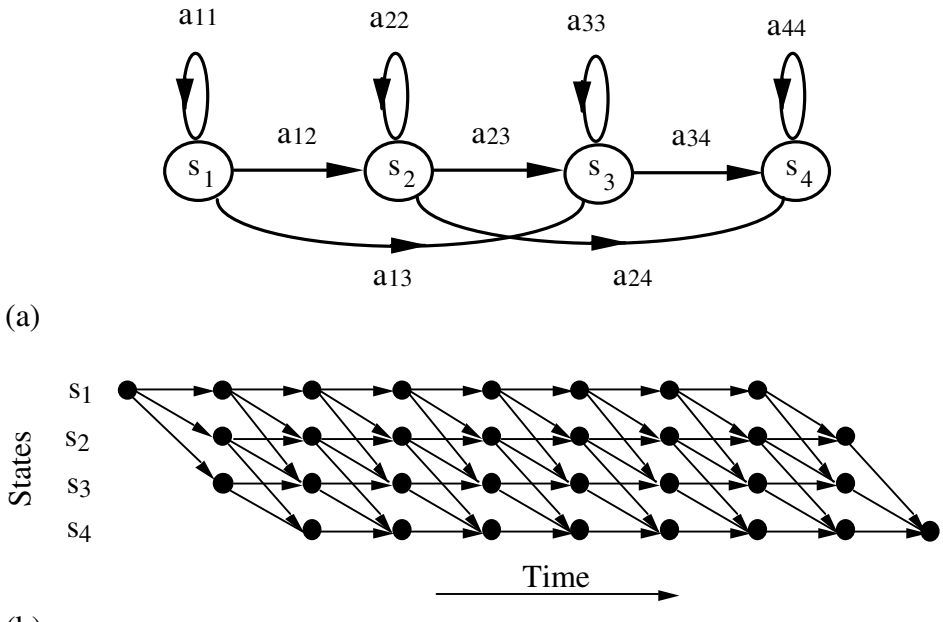


Figure 5.8 (a) A 4-state left–right HMM, and (b) its state–time trellis diagram.

5.2.6 State–Time Trellis Diagram

A state–time trellis diagram shows the HMM states together with all the different paths that can be taken through various states as time unfolds. Figure 5.8(a) and 5.8(b) illustrate a 4-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 an N-state HMM can reproduce N^T different realisations of the random process that it is trained to model.

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 $[\boldsymbol{X}]=[\boldsymbol{X}_k;\ k=0,\ ...,\ L-1]$, with each sequence $\boldsymbol{X}_k=[\boldsymbol{x}(t);\ t=0,\ ...,\ T_k-1]$ 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 $P_{\mathcal{M}|\boldsymbol{X}}(\mathcal{M}|\boldsymbol{X})$ 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

$$P_{\mathcal{M}|\boldsymbol{X}}(\mathcal{M}|\boldsymbol{X})=\frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})}f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})P_{\mathcal{M}}(\mathcal{M}) \tag{5.10}$$

where the denominator $f_{\boldsymbol{X}}(\boldsymbol{X})$ 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 $P_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$. The likelihood of an observation vector sequence $\boldsymbol{X}$ given a model $\mathcal{M}$ can be expressed as

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})=\sum_{\boldsymbol{s}} f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})P_{\boldsymbol{s}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}) \tag{5.11}$$

where $f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}(t)|s(t),\mathcal{M})$, 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

$$f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\mathbf{s},\mathcal{M})=f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(0)|s(0))f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(1)|s(1))\cdots f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.12}$$

where $s(t)$, the state at time t, can be one of N states, and $f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{X}(t)|s(t))$, a shorthand for $f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}(t)|s(t),\mathcal{M})$, 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

$$P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})=\pi_{s(0)}a_{s(0)s(1)}a_{s(1)s(2)}\cdots a_{s(T-2)s(T-1)} \tag{5.13}$$

Substituting Equations (5.12) and (5.13) in Equation (5.11) yields

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}\,|\,\mathcal{M}) = \sum_{\mathbf{s}} f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}\,|\,\boldsymbol{s},\mathcal{M}) P_{\boldsymbol{s}|\mathcal{M}}(\boldsymbol{s}\,|\,\mathcal{M})$$
$$= \sum_{\mathbf{s}} \pi_{s(0)} f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(0)|s(0)) a_{s(0)s(1)} f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(1)|s(1)) \cdots a_{s(T-2)s(T-1)} f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.14}$$

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 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 $f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$ 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), ..., $\boldsymbol{x}$(t)] and the state i at time t, of the model $\mathcal{M}$:

$$\alpha_t(i) = f_{\boldsymbol{X},S|\mathcal{M}}(\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(t), s(t) = i\,|\mathcal{M}) \tag{5.15}$$

The forward probability variable $\alpha_t(i)$ of Equation (5.15) can be expressed in a recursive form in terms of the forward probabilities at time t–1, $\alpha_{t-1}(i)$:

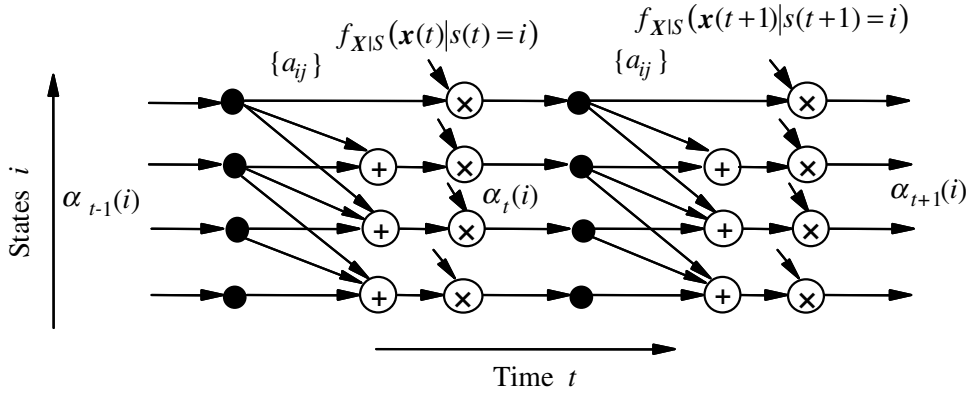


Figure 5.9 A network for computation of forward probabilities for a left-right HMM.

$$\begin{aligned}\alpha_t(i) &= f_{\boldsymbol{X},S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(t),s(t)=i|\mathcal{M}\big)\\ &= \left(\sum_{j=1}^{N} f_{\boldsymbol{X},S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(t-1),s(t-1)=j|\mathcal{M}\big)a_{ji}\right) f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}\big(\boldsymbol{x}(t)|s(t)=i,\mathcal{M}\big)\\ &= \sum_{j=1}^{N}\big(\alpha_{t-1}(j)\,a_{ji}\big) f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}\big(\boldsymbol{x}(t)|s(t)=i,\mathcal{M}\big)\end{aligned} \tag{5.16}$$

Figure 5.9 illustrates, a network for computation of the forward probabilities for the 4-state left–right HMM of Figure 5.8. The likelihood of an observation sequence $\boldsymbol{X}=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{x}(T-1)]$ given a model $\mathcal{M}$ can be expressed in terms of the forward probabilities as

$$\begin{aligned}f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(T-1)|\mathcal{M}) &= \sum_{i=1}^{N} f_{\boldsymbol{X},S|\mathcal{M}}(\boldsymbol{x}(0),\boldsymbol{x}(1),\ldots,\boldsymbol{x}(T-1),s(T-1)=i|\mathcal{M})\\ &= \sum_{i=1}^{N}\alpha_{T-1}(i)\end{aligned} \tag{5.17}$$

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 $[\boldsymbol{x}(t+1), \boldsymbol{x}(t+2), ..., \boldsymbol{x}(T-1)]$ as

$$\begin{aligned}\beta_t(i) &= f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(s(t)=i,\boldsymbol{x}(t+1),\boldsymbol{x}(t+2),\ldots,\boldsymbol{x}(T-1)|\mathcal{M}) \\ &= \sum_{j=1}^{N} a_{ij} f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(s(t+1)=j,\boldsymbol{x}(t+2),\boldsymbol{x}(t+3),\ldots,\boldsymbol{x}(T-1)) \\ &\quad \times f_{\boldsymbol{X}|\boldsymbol{S}}(\boldsymbol{x}(t+1)|s(t+1)=j,|\mathcal{M}) \\ &= \sum_{j=1}^{N} a_{ij}\beta_{t+1}(j) f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{x}(t+1)|s(t+1)=j,|\mathcal{M})\end{aligned} \tag{5.18}$$

In the next section, forward and backward probabilities are used to develop a method for the training of HMM parameters.

5.3.2 Baum–Welch Model Re-Estimation

The HMM training problem is the estimation of the model parameters $\mathcal{M}=(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{F})$ for a given data set. 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 $\{\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]\}$, with the objective of maximising $f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$, 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

$$\begin{aligned}\gamma_t(i,j) &= P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(s(t)=i,s(t+1)=j|\boldsymbol{X},\mathcal{M}) \\ &= \frac{f_{\boldsymbol{S},\boldsymbol{X}|\mathcal{M}}(s(t)=i,s(t+1)=j,\boldsymbol{X}|\mathcal{M})}{f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})} \\ &= \frac{\alpha_t(i)a_{ij}f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{x}(t+1)|s(t+1)=j,\mathcal{M})\beta_{t+1}(j)}{\sum_{i=1}^{N}\alpha_{T-1}(i)}\end{aligned} \tag{5.19}$$

where $f_{\boldsymbol{S},\boldsymbol{X}|\mathcal{M}}(s(t)=i,s(t+1)=j,\boldsymbol{X}|\mathcal{M})$ is the joint pdf of the states $s(t)$ and

$s(t+1)$ and the observation sequence $\boldsymbol{X}$, and $f_{\boldsymbol{X}|S}(\boldsymbol{x}(t+1)|s(t+1)=i)$ 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 $P_{\boldsymbol{X}|S}(\boldsymbol{x}(t+1)|s(t+1)=i)$. The posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is

$$\begin{aligned}\gamma_t(i) &= P_{S|\boldsymbol{X},\mathcal{M}}\left(s(t)=i\middle|\boldsymbol{X},\mathcal{M}\right) \\ &= \frac{f_{S,\boldsymbol{X}|\mathcal{M}}\left(s(t)=i,\boldsymbol{X}\middle|\mathcal{M}\right)}{f_{\boldsymbol{X}|\mathcal{M}}\left(\boldsymbol{X}\middle|\mathcal{M}\right)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.20}$$

Now the state transition probability a_{ij} can be interpreted as

$$a_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \tag{5.21}$$

From Equations (5.19)–(5.21), the state transition probability can be re-estimated as the ratio

$$\overline{a}_{ij} = \frac{\sum_{t=0}^{T-2}\gamma_t(i,j)}{\sum_{t=0}^{T-2}\gamma_t(i)} \tag{5.22}$$

Note that for an observation sequence $[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]$ 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

$$\overline{\pi}_i = \gamma_0(i) \tag{5.23}$$

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 $[\boldsymbol{\mu}_{i1},\boldsymbol{\mu}_{i2}, ..., \boldsymbol{\mu}_{iM}]$ 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 $f_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i)$ are substituted with probability mass functions $P_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i)$ defined as

$$P_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i)=P_{\boldsymbol{X}|S}(Q[\boldsymbol{x}(t)]|s(t)=i) \tag{5.24}$$

where the function $Q[\boldsymbol{x}(t)]$ quantises the observation vector $\boldsymbol{x}(t)$ to the nearest discrete vector in the set $[\boldsymbol{\mu}_{i1},\boldsymbol{\mu}_{i2}, ..., \boldsymbol{\mu}_{iM}]$. 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:

$$\begin{aligned}\overline{P}_{ik}(\boldsymbol{\mu}_{ik}) &= \frac{\text{expected number of times in state } i \text{ and observing } \boldsymbol{\mu}_{ik}}{\text{expected number of times in state } i} \\ &= \frac{\sum_{t\in \boldsymbol{x}(t)\rightarrow\boldsymbol{\mu}_{ik}}^{T-1}\gamma_t(i)}{\sum_{t=0}^{T-1}\gamma_t(i)}\end{aligned} \tag{5.25}$$

In Equation (5.25) the summation in the numerator is taken over those time instants t 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 $\boldsymbol{X}=[\boldsymbol{X}(0), \boldsymbol{X}(1), ..., \boldsymbol{X}(L-1)]$, where $\boldsymbol{X}(k)=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{x}(T_k-1)]$. The re-estimation formula can be averaged over the entire data set as

$$\hat{\pi}_i = \frac{1}{L}\sum_{l=0}^{L-1}\gamma_0^l(i) \tag{5.26}$$

$$\hat{a}_{ij} = \frac{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-2}\gamma_t^l(i,j)}{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-2}\gamma_t^l(i)} \tag{5.27}$$

and

$$\hat{P}_i(\boldsymbol{\mu}_{ik}) = \frac{\sum_{l=0}^{L-1}\sum_{t\in \boldsymbol{x}(t)\to\boldsymbol{\mu}_{ik}}^{T_l-1}\gamma_t^l(i)}{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-1}\gamma_t^l(i)} \tag{5.28}$$

The parameter estimates of Equations (5.26)–(5.28) can be used in further iterations of the estimation process until the model converges.

5.3.4 HMMs with Continuous Density Observation Models

In continuous density HMMs, continuous 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 a Gaussian pdf. A continuous *P*-variate Gaussian pdf for the state *i* of an HMM can be defined as

$$f_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i) = \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_i|^{1/2}}\exp\left\{[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]\right\} \tag{5.29}$$

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

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)\boldsymbol{x}(t)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.30}$$

Similarly, the covariance matrix is estimated as

$$\bar{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)(\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_i)(\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_i)^{\mathrm{T}}}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.31}$$

The proof that the Baum–Welch re-estimation algorithm leads to maximisation of the likelihood function $f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$ can be found in Baum.

5.3.5 HMMs with Mixture Gaussian pdfs

The modelling of the space of a signal process with a mixture of Gaussian pdfs is considered in Section 4.5. In HMMs with mixture Gaussian pdf state models, the signal space associated with the i^{th} state is modelled with a mixtures of M Gaussian densities as

$$f_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i) = \sum_{k=1}^{M} P_{ik}\, \mathcal{N}(\boldsymbol{x}(t), \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \tag{5.32}$$

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), ..., $\boldsymbol{x}$(T–1)], can be expressed as

$$\begin{aligned}\gamma_t(i,j) &= P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(s(t)=i, s(t+1)=j \mid \boldsymbol{X}, \mathcal{M}) \\ &= \frac{\alpha_t(i)a_{ij}\left[\sum_{k=1}^{M} P_{jk}\mathcal{N}(\boldsymbol{x}(t+1), \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})\right]\beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_{T-1}(i)}\end{aligned} \tag{5.33}$$

and the posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is given by

$$\begin{aligned}\gamma_t(i) &= P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}\left(s(t)=i\middle|\boldsymbol{X},\mathcal{M}\right)\\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.34}$$

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

$$\begin{aligned}\zeta_t(i,k) &= P_{S,K|\boldsymbol{X},\mathcal{M}}\left(s(t)=i,m(t)=k\middle|\boldsymbol{X},\mathcal{M}\right)\\ &= \frac{\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji}P_{ik}\mathcal{N}\left(\boldsymbol{x}(t),\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik}\right)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}\end{aligned} \tag{5.35}$$

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 of Signals Using Hidden Markov Models

Hidden Markov models are used in applications such as speech recognition, image recognition and signal restoration, and for the decoding of the underlying states of a signal. For example, in speech recognition, HMMs are trained to model the statistical variations of the acoustic realisations of the words in a vocabulary of say size V words. In the word recognition phase, an utterance is classified and labelled with the most likely of the V+1 candidate HMMs (including an HMM for silence) as illustrated in Figure 5.10. In Chapter 12 on the modelling and detection of impulsive noise, a binary–state HMM is used to model the impulsive noise process.

Consider the decoding of an unlabelled sequence of T signal vectors $\boldsymbol{X}=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{X}(T-1)]$ given a set of V candidate HMMs $[\mathcal{M}_1, ..., \mathcal{M}_V]$. The probability score for the observation vector sequence $\boldsymbol{X}$ and the model $\mathcal{M}_k$ can be calculated as the likelihood:

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_k) = \sum_{\mathbf{s}} \pi_{s(0)} f_{\boldsymbol{X}|S}(\boldsymbol{x}(0)|s(0)) a_{s(0)s(1)} f_{\boldsymbol{X}|S}(\boldsymbol{x}(1)|s(1)) \cdots a_{s(T-2)s(T-1)} f_{\boldsymbol{X}|S}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.39}$$

where the likelihood of the observation sequence $\boldsymbol{X}$ is summed over all possible state sequences of the model $\mathcal{M}$. Equation (5.39) can be efficiently calculated using the forward–backward method described in Section 5.3.1. The observation sequence $\boldsymbol{X}$ is labelled with the HMM that scores the highest likelihood as

$$Label(\boldsymbol{X}) = \arg\max_{k} \left(f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_k) \right), \qquad k=1, ..., V+1 \tag{5.40}$$

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

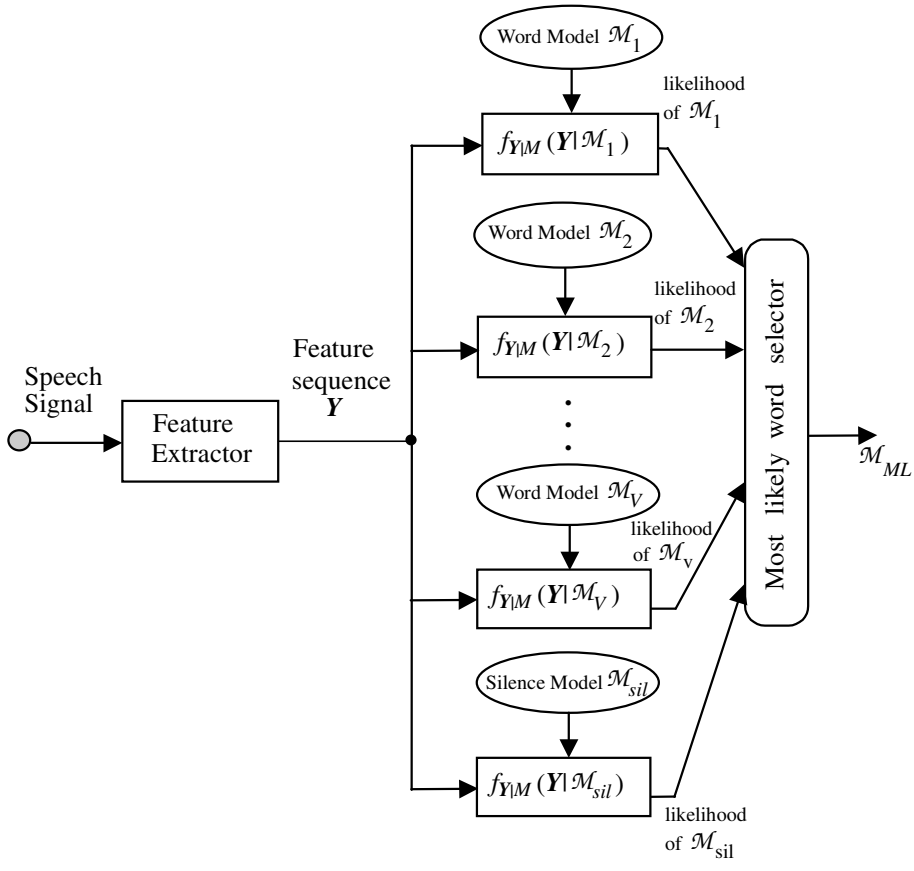


Figure 5.10 Illustration of the use of HMMs in speech recognition.

model $\mathcal{M}_k$, instead of being summed over all sequences, so Equation (5.40) becomes

$$Label(\boldsymbol{X}) = \arg\max_k \left[\max_{\boldsymbol{s}} f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(\boldsymbol{X},\boldsymbol{s}|\mathcal{M}_k) \right] \tag{5.41}$$

In Section 5.5, on the use of HMMs for noise reduction, the most likely state sequence is used to obtain the maximum-likelihood estimate of the underlying statistics of the signal process.

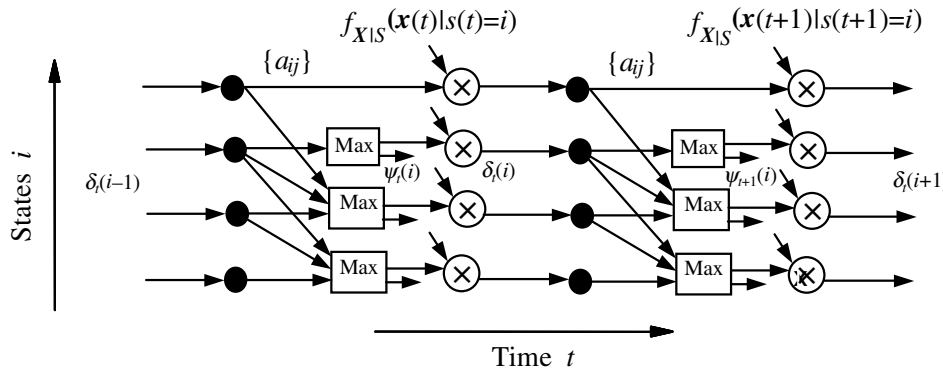


Figure 5.11 A network illustration of the Viterbi algorithm.

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 $\boldsymbol{s}^{MAP}$ of a model $\mathcal{M}$ given an observation signal sequence $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]$ is obtained as

$$\begin{aligned} \boldsymbol{s}^{MAP} &= \arg\max_{\boldsymbol{s}} f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(\boldsymbol{X},\boldsymbol{s}|\mathcal{M}) \\ &= \arg\max_{\boldsymbol{s}} \left(f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M}) P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}) \right) \end{aligned} \tag{5.42}$$

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 next section.

For an N-state HMM and an observation sequence of length T, there are altogether N^T state sequences. Even for moderate values of N and T say (N=6 and T=30), an exhaustive search of the state–time trellis for the best state sequence is a computationally prohibitive exercise. The Viterbi algorithm is an efficient method for the estimation of the most likely state sequence of an HMM. In a state–time trellis diagram, such as Figure 5.8, the number of paths diverging from each state of a trellis can grow exponentially by a factor of N at successive time instants. The Viterbi

method prunes the trellis by selecting the most likely path to each state. At each time instant t, for each state i, the algorithm selects the most probable path to state i and prunes out the less likely branches. This procedure ensures that at any time instant, only a single path *survives* into each state of the trellis.

For each time instant t and for each state i, the algorithm keeps a record of the state j from which the maximum-likelihood path branched into i, and also records the cumulative probability of the most likely path into state i at time t. The Viterbi algorithm is given on the next page, and Figure 5.11 gives a network illustration of the algorithm.

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, …, 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, …, T–1

For states i = 1, …, N

$$\delta_t(i)=\max_j[\delta_{t-1}(j)a_{ji}]f_i(\boldsymbol{x}(t))$$
$$\psi_t(i)=\arg\max_j[\delta_{t-1}(j)a_{ji}]$$

Step 3: *Termination,* retrieve the most likely final state

$$s^{MAP}(T-1)=\arg\max_i[\delta_{T-1}(i)]$$
$$Prob_{\max}=\max_i[\delta_{T-1}(i)]$$

Step 4: *Backtracking* through the most likely state sequence:

For t = T–2, …, 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 $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]$ 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 HMM-Based Estimation of Signals in Noise

In this section, and the following two sections, we consider the use of HMMs for estimation of a signal $\boldsymbol{x}(t)$ 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

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}(t)|\boldsymbol{y}(t)) &= \frac{f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}(t)|\boldsymbol{x}(t)) f_{\boldsymbol{X}}(\boldsymbol{x}(t))}{f_{\boldsymbol{Y}}(\boldsymbol{y}(t))} \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y}(t))} f_{\boldsymbol{N}}(\boldsymbol{y}(t) - \boldsymbol{x}(t)) f_{\boldsymbol{X}}(\boldsymbol{x}(t)) \end{aligned} \tag{5.44}$$

For a given observation, $f_{\boldsymbol{Y}}(\boldsymbol{y}(t))$ is a constant, and the maximum a posteriori (MAP) estimate is obtained as

$$\hat{\boldsymbol{x}}^{MAP}(t) = \arg\max_{\boldsymbol{x}(t)} f_{\boldsymbol{N}}(\boldsymbol{y}(t) - \boldsymbol{x}(t)) f_{\boldsymbol{X}}(\boldsymbol{x}(t)) \tag{5.45}$$

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 subprocesses. Now assume that we have an N_s-state HMM $\mathcal{M}$ for the signal, and another N_n-state HMM η for the noise. For signal estimation, we need estimates of the underlying state sequences of the signal and the noise processes. For an observation sequence of length T, there are N_s^T possible signal state sequences and N_n^T possible noise state sequences that could have generated the noisy signal. Since it is assumed that the signal and noise are uncorrelated, each signal state may be observed in any noisy state; therefore the number of noisy signal states is on the order of $N_s^T \times N_n^T$.

Given an observation sequence $\boldsymbol{Y}=[\boldsymbol{y}(0), \boldsymbol{y}(1), ..., \boldsymbol{y}(T-1)]$, the most probable state sequences of the signal and the noise HMMs maybe expressed as

$$\boldsymbol{s}_{\text{signal}}^{MAP} = \arg\max_{\boldsymbol{s}_{\text{signal}}} \left(\max_{\boldsymbol{s}_{\text{noise}}} f_{\boldsymbol{Y}}\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} \middle| \mathcal{M}, \eta\right) \right) \tag{5.46}$$

and

$$\boldsymbol{s}_{\text{noise}}^{MAP} = \arg\max_{\boldsymbol{s}_{\text{noise}}} \left(\max_{\boldsymbol{s}_{\text{signal}}} f_{\boldsymbol{Y}}\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} \middle| \mathcal{M}, \eta\right) \right) \tag{5.47}$$

Given the state sequence estimates for the signal and the noise models, the MAP estimation Equation (5.45) becomes

$$\hat{\boldsymbol{x}}^{MAP}(t) = \arg\max_{\boldsymbol{x}} \left(f_{\boldsymbol{N}|\boldsymbol{S},\eta}\left(\boldsymbol{y}(t) - \boldsymbol{x}(t) \middle| \boldsymbol{s}_{\text{noise}}^{MAP}, \eta\right) f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}\left(\boldsymbol{x}(t) \middle| \boldsymbol{s}_{\text{signal}}^{MAP}, \mathcal{M}\right) \right) \tag{5.48}$$

Implementation of Equations (5.46)–(5.48) is computationally prohibitive. In Sections 5.6 and 5.7, we consider some practical methods for the estimation of signal in noise.

Example 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 $\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{\mu}_{\boldsymbol{x}_0},\boldsymbol{\Sigma}_{\boldsymbol{x}_0\boldsymbol{x}_0})$, and $s(t)=1$ indicates that the signal is from the state S_1 with a Gaussian pdf of $\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{\mu}_{\boldsymbol{x}_1},\boldsymbol{\Sigma}_{\boldsymbol{x}_1\boldsymbol{x}_1})$. 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

$$\boldsymbol{s}_{\text{signal}}^{MAP}=\arg\max_{\boldsymbol{s}}\left[f_{\boldsymbol{Y}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{s},\mathcal{M})P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})\right] \tag{5.50}$$

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

$$f_{\boldsymbol{Y}|s_0}(\boldsymbol{y}(t)|s_0)=\mathcal{N}\big(\boldsymbol{y}(t),(\boldsymbol{\mu}_{\boldsymbol{x}_0}+\boldsymbol{\mu}_{\boldsymbol{n}}),(\boldsymbol{\Sigma}_{\boldsymbol{x}_0\boldsymbol{x}_0}+\boldsymbol{\Sigma}_{\boldsymbol{nn}})\big) \tag{5.51}$$

and

$$f_{\boldsymbol{Y}|s_1}(\boldsymbol{y}(t)|s_1)=\mathcal{N}\big(\boldsymbol{y}(t),(\boldsymbol{\mu}_{\boldsymbol{x}_1}+\boldsymbol{\mu}_{\boldsymbol{n}}),(\boldsymbol{\Sigma}_{\boldsymbol{x}_1\boldsymbol{x}_1}+\boldsymbol{\Sigma}_{\boldsymbol{nn}})\big) \tag{5.52}$$

where $\mathcal{N}\big(\boldsymbol{y}(t),\boldsymbol{\mu},\boldsymbol{\Sigma}\big)$ denotes a Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The MAP signal estimate, given a state sequence estimate $\boldsymbol{s}^{MAP}$, is obtained from

$$\hat{\boldsymbol{x}}^{MAP}(t)=\arg\max_{\boldsymbol{x}}\left[f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}\big(\boldsymbol{x}(t)\big|\boldsymbol{s}^{MAP},\mathcal{M}\big)f_{\boldsymbol{N}}(\boldsymbol{y}(t)-\boldsymbol{x}(t))\right] \tag{5.53}$$

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 $[s(t)]$.

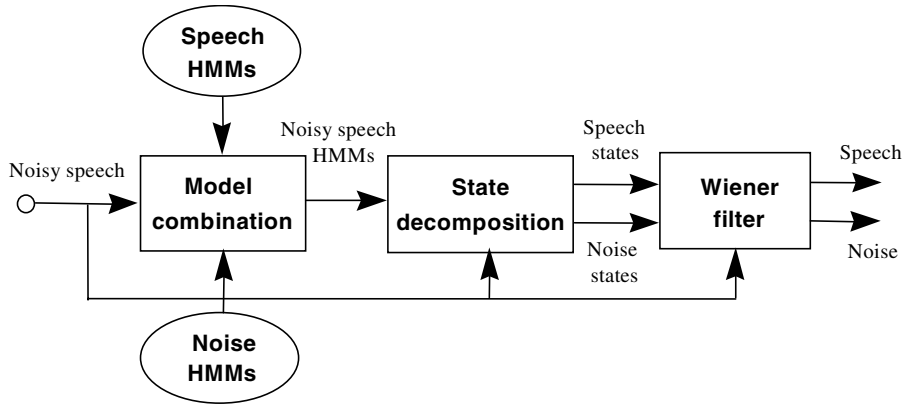


Figure 5.12 Outline configuration of HMM-based noisy speech recognition and enhancement.

5.6 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 5.12 illustrates the outline of an HMM-based noisy speech recognition and enhancement system. The system performs the following functions:

(1) combination of the speech and noise HMMs to form the noisy speech HMMs;
(2) estimation of the best combined noisy speech model given the current noisy speech input;
(3) state decomposition, i.e. the separation of speech and noise states given noisy speech states;
(4) state-based Wiener filtering using the estimates of speech and noise states.

5.6.1 Hidden Markov Model Combination

The performance of HMMs trained on clean signals deteriorates rapidly in the presence of noise, since noise causes a mismatch between the clean HMMs and the noisy signals. The noise-induced mismatch can be reduced: either by filtering the noise from the signal (for example using the Wiener filtering and the spectral subtraction methods described in Chapters 6 and 11) or by combining the noise and the signal models to model the noisy

signal. The model combination method 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 5.13 illustrates the combination of a 4-state left–right HMM of a speech signal with a 2-state ergodic HMM of noise. Assuming that speech and noise are independent processes, each speech state must be combined with every possible noise state to give the noisy speech model. It is assumed that the noise process only affects the mean vectors and the covariance matrices of the speech model; hence the transition probabilities of the speech model are not modified.

5.6.2 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:

(a) 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.
(b) Obtain the ML state sequence of from the ML combined model.
(c) Extract the signal and noise states from the ML state sequence of the ML combined noisy signal model.

The ML state sequences provide the probability density functions for the signal and noise processes. The ML estimates of the speech and noise pdfs

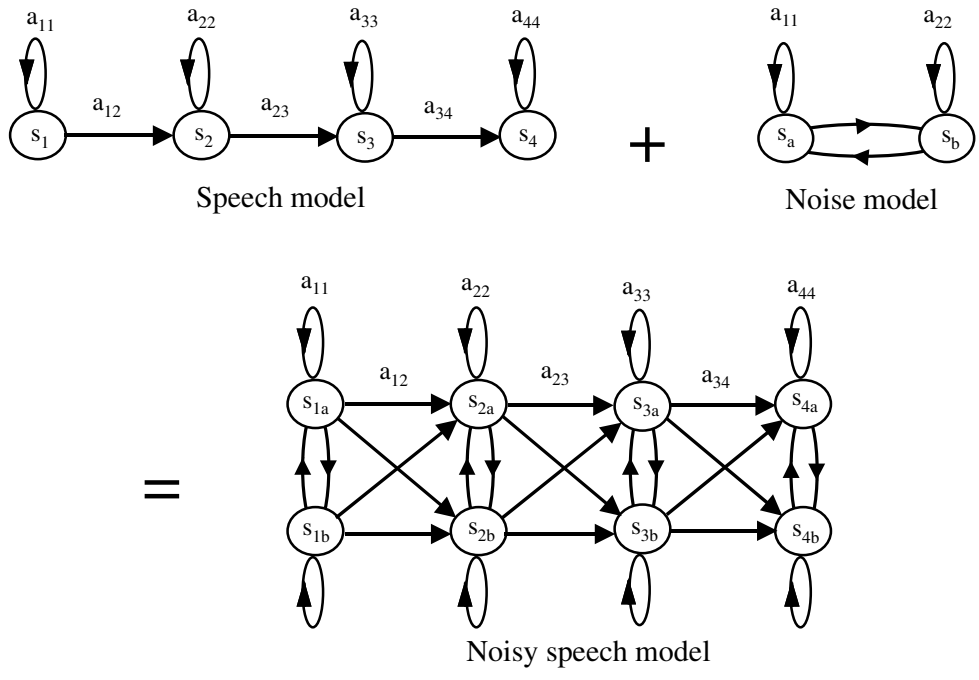


Figure 5.13 Outline configuration of HMM-based noisy speech recognition and enhancement. S_{ij} is a combination of the state *i* of speech with the state *j* of noise.

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.

5.7 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 :

$$\boldsymbol{w} = (\boldsymbol{R}_{\boldsymbol{xx}} + \boldsymbol{R}_{\boldsymbol{nn}})^{-1}\boldsymbol{r}_{\boldsymbol{xx}} \tag{5.55}$$

and

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{5.56}$$

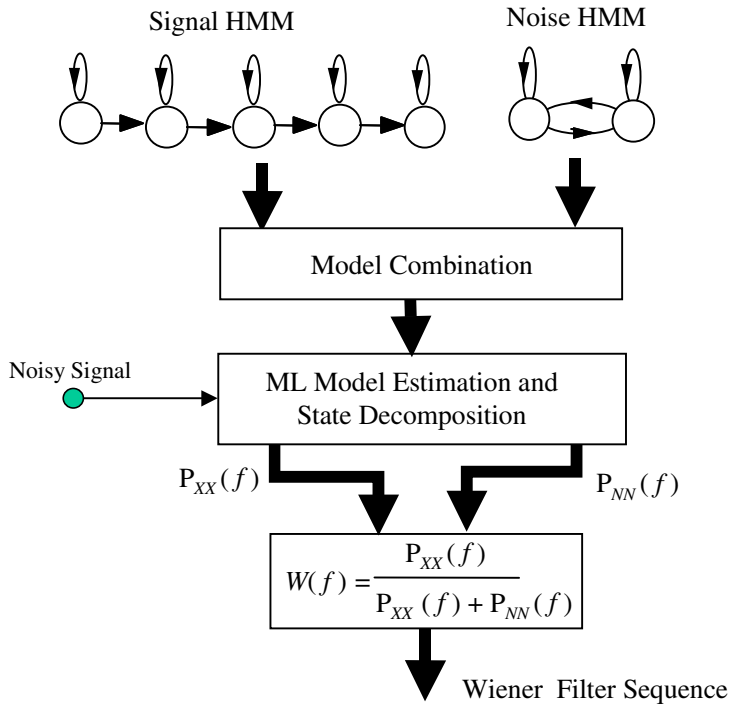


Figure 5.14 Illustrations of HMMs with state-dependent Wiener filters.

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 (5.56), 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 5.14 illustrates an implementation of HMM-based state-dependent Wiener filters. To implement the state-dependent Wiener filter, we need an estimate of the state sequences for the signal and the noise. In practice, for signals such as speech there are a number of HMMs; one HMM per word, phoneme, or any other elementary unit of the signal. In such cases it is necessary to classify the signal, so that the state-based Wiener filters are derived from the most likely HMM. Furthermore the noise process can also be modelled by an HMM. Assuming that there are V HMMs $\{\mathcal{M}_1, ..., \mathcal{M}_V\}$ 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.

5.7.1 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 5.15. Figure 5.16 shows a two-state HMM of the impulsive noise sequence where the state S_0 models the "off" periods between the impulses and the state S_1 models an impulse. In cases where each impulse has a well-defined temporal structure, it may be beneficial to use a multistate HMM to model the pulse itself. HMMs are used in Chapter 12 for modelling impulsive noise, and in Chapter 15 for channel equalisation.

5.8 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.

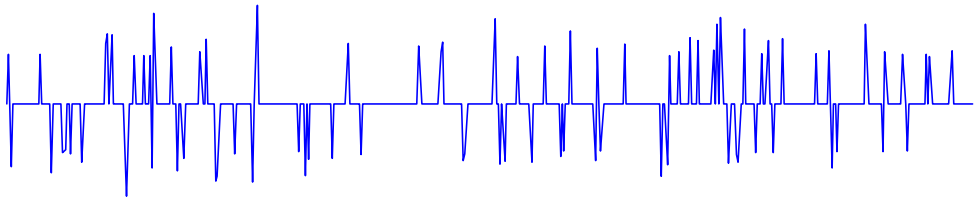


Figure 5.15 Impulsive noise.

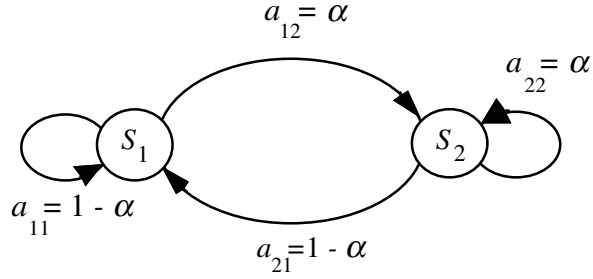


Figure 5.16 A binary-state model of an impulsive noise process.

In Section 5.3, we studied the Baum–Welch method for the training of the parameters of an HMM to model a given data set, and derived the forward–backward method for efficient calculation of the likelihood of an HMM given an observation signal. In Section 5.4, we considered the use of HMMs in signal classification and in the decoding of the underlying state sequence of a signal. The Viterbi algorithm is a computationally efficient method for estimation of the most likely sequence of an HMM. Given an unlabelled observation signal, the decoding of the underlying state sequence and the labelling of the observation with one of number of candidate HMMs are accomplished using the Viterbi method. In Section 5.5, we considered the use of HMMs for MAP estimation of a signal observed in noise, and considered the use of HMMs in implementation of state-based Wiener filter sequence.

Bibliography

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, pp. 40–43.

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**, pp. 179–190.

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**, pp. 360-363.

BAUM L.E. and PETRIE T. (1966) Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Ann. Math. Stat. **37**, pp. 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**, pp. 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)**, pp. 1846-1856, Dec.

FORNEY G.D. (1973) The Viterbi Algorithm. Proc. IEEE, **61**, pp. 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, pp. 233–235.

GALES M.J.F. and YOUNG S.J. (1993) HMM Recognition in Noise using Parallel Model Combination. Eurospeech-93, pp. 837–840.

HUANG X.D., ARIKI Y. and JACK M.A. (1990) Hidden Markov Models for Speech Recognition. Edinburgh University Press, Edinburgh.

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, pp. 639–642.

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, Amesterdam.

JELINEK F, (1976) Continuous Speech Recognition by Statistical Methods. Proc. of IEEE, **64**, pp. 532–555.

JUANG B.H. (1985) Maximum-Likelihood Estimation for Mixture Multi-Variate Stochastic Observations of Markov Chain. AT&T Bell laboratories Tech J., 64, pp. 1235–1249.

JUANG B.H. (1984) On the Hidden Markov Model and Dynamic Time

Warping for Speech Recognition- A unified Overview. AT&T Technical J., **63**, pp. 1213–1243.

KULLBACK S. and LEIBLER R.A. (1951) On Information and Sufficiency. Ann. Math. Stat., **22**, pp. 79–85.

LEE K.F. (1989) Automatic Speech Recognition: the Development of SPHINX System. MA: Kluwer Academic Publishers, Boston.

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-28**, pp. 729–735.

MARKOV A.A. (1913) An Example of Statistical Investigation in the text of *Eugen Onyegin* Illustrating Coupling of Tests in Chains. Proc. Acad. Sci. St Petersburg VI Ser., **7**, pp. 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**, pp. 97–115.

RABINER L.R. and JUANG B.H. (1986) An Introduction to Hidden Markov Models. IEEE ASSP. Magazine, pp. 4–15.

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 Technical Journal, **64**, pp. 1211-1235.

RABINER L.R. and JUANG B.H. (1993) Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

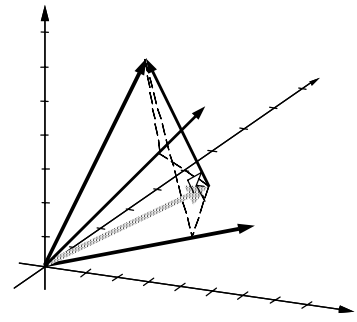
YOUNG S.J. (1999), HTK: Hidden Markov Model Tool Kit. Cambridge University Engineering Department.

VARGA A. and MOORE R.K., Hidden Markov Model Decomposition of Speech and Noise. in Proc. IEEE Int., Conf. on Acoust., Speech, Signal Processing, 1990, pp. 845–848

VITERBI A.J. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Trans. on Information theory, **IT-13**, pp. 260–269.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)



6

WIENER FILTERS

Wiener theory, formulated by Norbert Wiener, forms the foundation of data-dependent linear least square error filters. Wiener filters play a central role in a wide range of applications such as linear prediction, echo cancellation, signal restoration, channel equalisation and system identification. The coefficients of a Wiener filter are calculated to minimise the average squared distance between the filter output and a desired signal. In its basic form, the Wiener theory assumes that the signals are stationary processes. However, if the filter coefficients are periodically recalculated for every block of N signal samples then the filter adapts itself to the average characteristics of the signals within the blocks and becomes block-adaptive. A block-adaptive (or segment adaptive) filter can be used for signals such as speech and image that may be considered almost stationary over a relatively small block of samples. In this chapter, we study Wiener filter theory, and consider alternative methods of formulation of the Wiener filter problem. We consider the application of Wiener filters in channel equalisation, time-delay estimation and additive noise reduction. A case study of the frequency response of a Wiener filter, for additive noise reduction, provides useful insight into the operation of the filter. We also deal with some implementation issues of Wiener filters.

6.1 Wiener Filters: Least Square Error Estimation

Wiener formulated the continuous-time, least mean square error, estimation problem in his classic work on interpolation, extrapolation and smoothing of time series (Wiener 1949). The extension of the Wiener theory from continuous time to discrete time is simple, and of more practical use for implementation on digital signal processors. A Wiener filter can be an infinite-duration impulse response (IIR) filter or a finite-duration impulse response (FIR) filter. In general, the formulation of an IIR Wiener filter results in a set of non-linear equations, whereas the formulation of an FIR Wiener filter results in a set of linear equations and has a closed-form solution. In this chapter, we consider FIR Wiener filters, since they are relatively simple to compute, inherently stable and more practical. The main drawback of FIR filters compared with IIR filters is that they may need a large number of coefficients to approximate a desired response.
Figure 6.1 illustrates a Wiener filter represented by the coefficient vector $\boldsymbol{w}$. The filter takes as the input a signal $y(m)$, and produces an output signal $\hat{x}(m)$, where $\hat{x}(m)$ is the least mean square error estimate of a 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, $\boldsymbol{y}^{\mathrm{T}}=[y(m), y(m-1), ..., y(m-P-1)]$ is the filter input signal, and the parameter vector $\boldsymbol{w}^{\mathrm{T}}=[w_0, w_1, ..., w_{P-1}]$ is the Wiener filter coefficient vector. In Equation (6.1), the filtering operation is expressed in two alternative and equivalent forms of a convolutional sum and an inner vector product. The Wiener filter error signal, $e(m)$ is defined as the difference between the desired signal $x(m)$ and the filter output signal $\hat{x}(m)$:

$$\begin{aligned}e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{y}\end{aligned} \tag{6.2}$$

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 $\boldsymbol{w}$.

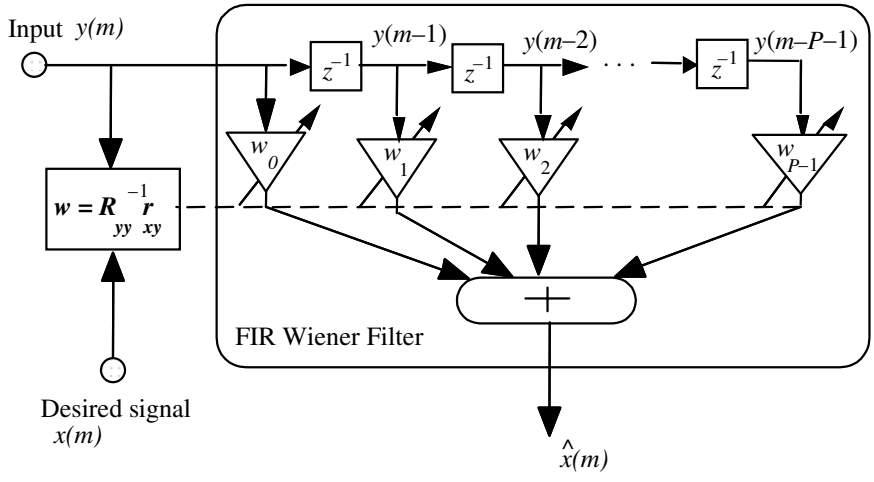


Figure 6.1 Illustration of a Wiener filter structure.

To explore the relation between the filter coefficient vector $\boldsymbol{w}$ and the error signal $e(m)$ we expand Equation (6.2) for N samples of the signals $x(m)$ and $y(m)$:

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} y(0) & y(-1) & y(-2) & \dots & y(1-P) \\ y(1) & y(0) & y(-1) & \dots & y(2-P) \\ y(2) & y(1) & y(0) & \dots & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & y(N-3) & \dots & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} \tag{6.3}$$

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.

In Equation (6.3), if the number of signal samples is equal to the number of filter coefficients $N=P$, then we have a square matrix equation, and there is a unique filter solution $\boldsymbol{w}$, with a zero estimation 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 is insufficient to obtain a unique solution for the filter coefficients, in this case there are an infinite number of solutions with zero estimation error, and the matrix equation is said to be *underdetermined*. In practice, the number of signal samples is much larger than the filter length $N>P$; in this case, the matrix equation is said to be *overdetermined* 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 average absolute value of error $\mathcal{E}[|e(m)|]$, or the mean square error $\mathcal{E}[e^2(m)]$, where $\mathcal{E}[.]$ is the expectation 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 followings, 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 $\boldsymbol{w}$. From Equation (6.2), the mean square estimation error is given by

$$\begin{aligned}\mathcal{E}[e^2(m)] &= \mathcal{E}[(x(m) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{y})^2] \\ &= \mathcal{E}[x^2(m)] - 2\boldsymbol{w}^{\mathrm{T}}\mathcal{E}[\boldsymbol{y}x(m)] + \boldsymbol{w}^{\mathrm{T}}\mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}]\boldsymbol{w} \\ &= r_{xx}(0) - 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{r}_{yx} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w}\end{aligned} \tag{6.5}$$

where $\boldsymbol{R}_{yy} = \mathcal{E}[\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)]$ is the autocorrelation matrix of the input signal and $\boldsymbol{r}_{xy} = \mathcal{E}[x(m)\boldsymbol{y}(m)]$ 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 only two coefficients (w_0, w_1), the mean square error function is a

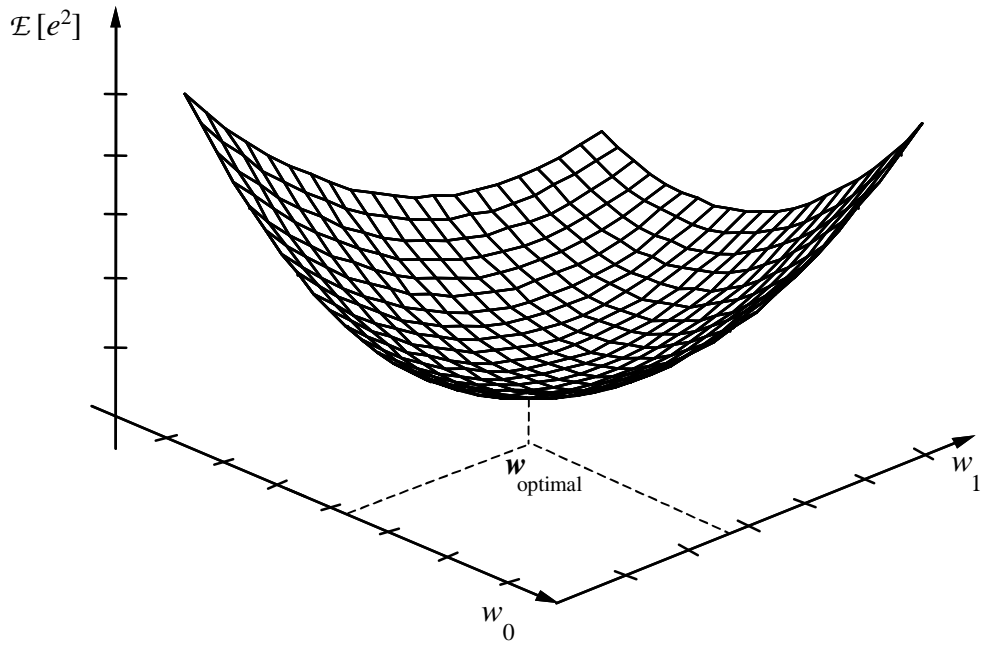


Figure 6.2 Mean square error surface for a two-tap FIR filter.

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 optimal 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

$$\frac{\partial}{\partial \boldsymbol{w}} = \left[\frac{\partial}{\partial w_0}, \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \ldots, \frac{\partial}{\partial w_{P-1}}\right]^{\mathrm{T}} \tag{6.8}$$

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

$$\begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-1} \end{pmatrix} = \begin{pmatrix} r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \dots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \dots & r_{yy}(P-2) \\ r_{yy}(2) & \mathrm{r}_{yy}(1) & \mathrm{r}_{yy}(0) & \dots & \mathrm{r}_{yy}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & r_{yy}(P-3) & \dots & r_{yy}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{yx}(0) \\ r_{yx}(1) \\ r_{yx}(2) \\ \vdots \\ r_{yx}(P-1) \end{pmatrix} \tag{6.11}$$

From Equation (6.11), the calculation of the Wiener filter coefficients requires the autocorrelation matrix of the input signal and the cross-correlation vector of the input and the desired signals.

In statistical signal processing theory, the correlation values of a random process are obtained as the averages taken across the ensemble of different realisations of the process as described in Chapter 3. However in many practical situations there are only one or two finite-duration realisations of the signals $x(m)$ and $y(m)$. 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 constant 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 methods.

6.2 Block-Data Formulation of the Wiener Filter

In this section we consider an alternative formulation of a Wiener filter for a block of N samples of the input signal [$y(0)$, $y(1)$, ..., $y(N–1)$] and the desired signal [$x(0)$, $x(1)$, ..., $x(N–1)$]. The set of N linear equations describing the Wiener filter input/output relation can be written in matrix form as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = \begin{pmatrix} y(0) & y(-1) & y(-2) & \cdots & y(2-P) & y(1-P) \\ y(1) & y(0) & y(-1) & \cdots & y(3-P) & y(2-P) \\ y(2) & y(1) & y(0) & \cdots & y(4-P) & y(3-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(N-2) & y(N-3) & y(N-4) & \cdots & y(N-P) & y(N-1-P) \\ y(N-1) & y(N-2) & y(N-3) & \cdots & y(N+1-P) & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{P-2} \\ w_{P-1} \end{pmatrix} \tag{6.13}$$

Equation (6.13) can be rewritten in compact matrix notation as

$$\hat{\boldsymbol{x}} = \boldsymbol{Y}\boldsymbol{w} \tag{6.14}$$

The Wiener filter error is the difference between the desired signal and the filter output defined as

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{x} - \hat{\boldsymbol{x}} \\ &= \boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w} \end{aligned} \tag{6.15}$$

The energy of the error vector, that is the sum of the squared elements of the error vector, is given by the inner vector product as

$$\begin{aligned} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} &= (\boldsymbol{x}-\boldsymbol{Y}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{Y}\boldsymbol{w}) \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{w} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{w} \end{aligned} \tag{6.16}$$

The gradient of the squared error function with respect to the Wiener filter coefficients is obtained by differentiating Equation (6.16):

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{w}} = -2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Y} + 2\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} \tag{6.17}$$

The Wiener filter coefficients are obtained by setting the gradient of the squared error function of Equation (6.17) to zero, this yields

$$\left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)\boldsymbol{w}=\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} \tag{6.18}$$

or

$$\boldsymbol{w}=\left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x} \tag{6.19}$$

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. Theoretically, the Wiener filter is obtained from minimisation of the squared error across the ensemble of different realisations of a process as described in the previous section. For a correlation-ergodic process, as the signal length N approaches infinity the block-data Wiener filter of Equation (6.19) approaches the Wiener filter of Equation (6.10):

$$\lim_{N\to\infty}\left[\boldsymbol{w}=\left(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{x}\right]=\boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{xy} \tag{6.20}$$

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.

6.2.1 QR Decomposition of the Least Square Error Equation

An efficient and robust method for solving the least square error Equation (6.19) is the QR decomposition (QRD) method. In this method, the $N\times P$ signal matrix $\boldsymbol{Y}$ is decomposed into the product of an $N\times N$ orthonormal matrix $\boldsymbol{Q}$ and a $P\times P$ upper-triangular matrix $\mathcal{R}$ as

$$\boldsymbol{Q}\boldsymbol{Y}=\begin{pmatrix}\mathcal{R}\\ \boldsymbol{0}\end{pmatrix} \tag{6.21}$$

where $\mathbf{0}$ is the $(N-P)\times P$ null matrix, $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}=\mathbf{I}$, and the upper-triangular matrix $\mathcal{R}$ is of the form

$$\mathcal{R}=\begin{pmatrix} r_{00} & r_{01} & r_{02} & r_{03} & \cdots & r_{0P-1} \\ 0 & r_{11} & r_{12} & r_{13} & \cdots & r_{1P-1} \\ 0 & 0 & r_{22} & r_{23} & \cdots & r_{2P-1} \\ 0 & 0 & 0 & r_{33} & \cdots & r_{3P-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_{P-1P-1} \end{pmatrix} \tag{6.22}$$

Substitution of Equation (6.21) in Equation (6.18) yields

$$\begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix}^{\mathrm{T}} \boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}} \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \boldsymbol{w} = \begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix}^{\mathrm{T}} \boldsymbol{Q}\boldsymbol{x} \tag{6.23}$$

From Equation (6.23) we have

$$\begin{pmatrix} \mathcal{R} \\ \mathbf{0} \end{pmatrix} \boldsymbol{w} = \boldsymbol{Q}\boldsymbol{x} \tag{6.24}$$

From Equation (6.24) we have

$$\mathcal{R}\boldsymbol{w} = \boldsymbol{x}_Q \tag{6.25}$$

where the vector $\boldsymbol{x}_Q$ on the right hand side of Equation (6.25) is composed of the first P elements of the product $\boldsymbol{Q}\boldsymbol{x}$. Since the matrix $\mathcal{R}$ is upper-triangular, the coefficients of the least square error filter can be obtained easily through a process of back substitution from Equation (6.25), 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 $\mathcal{R}$. The decomposition of a matrix into QR matrices can be achieved using a number of methods, including the Gram-Schmidt orthogonalisation method, the Householder method and the Givens rotation method.

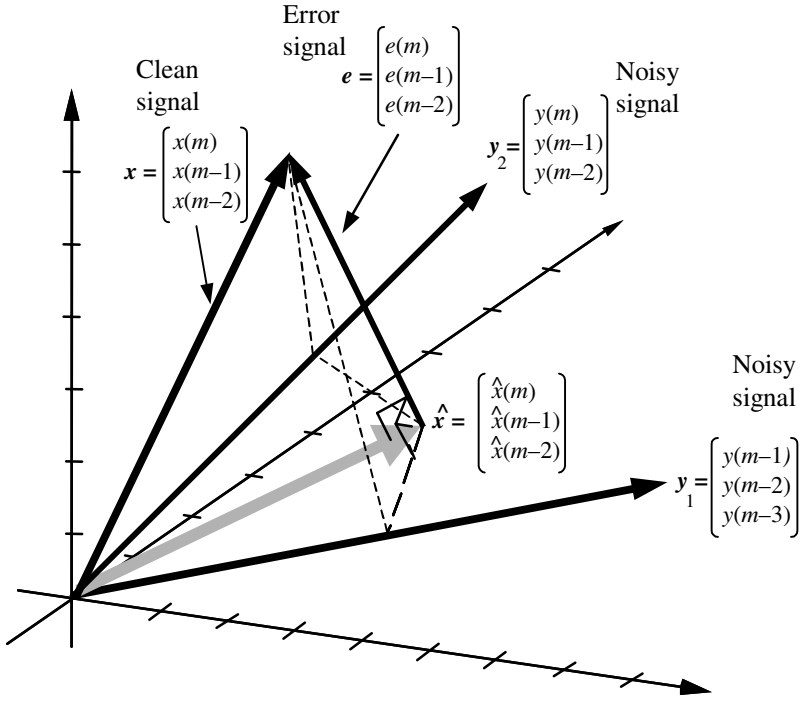


Figure 6.3 The least square error projection of a desired signal vector $\boldsymbol{x}$ onto a plane containing the input signal vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ is the perpendicular projection of $\boldsymbol{x}$ shown as the shaded vector.

6.3 Interpretation of Wiener Filters as Projection in Vector Space

In this section, we consider an alternative formulation of Wiener filters where the least square error estimate is visualized as the perpendicular minimum distance *projection* of the desired signal vector onto the vector space of the input signal. A vector space is the collection of an infinite number of vectors that can be obtained from linear combinations of a number of independent vectors.

In order to develop a vector space interpretation of the least square error estimation problem, we rewrite the matrix Equation (6.11) and express the filter output vector $\hat{\boldsymbol{x}}$ as a linear weighted combination of the column vectors of the input signal matrix as

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \hat{x}(2) \\ \vdots \\ \hat{x}(N-2) \\ \hat{x}(N-1) \end{pmatrix} = w_0 \begin{pmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N-2) \\ y(N-1) \end{pmatrix} + w_1 \begin{pmatrix} y(-1) \\ y(0) \\ y(1) \\ \vdots \\ y(N-3) \\ y(N-2) \end{pmatrix} + \cdots + w_{P-1} \begin{pmatrix} y(1-P) \\ y(2-P) \\ y(3-P) \\ \vdots \\ y(N-1-P) \\ y(N-P) \end{pmatrix} \tag{6.26}$$

In compact notation, Equation (6.26) may be written as

$$\hat{\boldsymbol{x}} = w_0 \boldsymbol{y}_0 + w_1 \boldsymbol{y}_1 + \cdots + w_{P-1} \boldsymbol{y}_{P-1} \tag{6.27}$$

In Equation (6.27) the signal estimate $\hat{\boldsymbol{x}}$ is 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 P N-dimensional input signal vectors $[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{P-1}]$ in Equation (6.27) define the *basis* vectors for a subspace in an N-dimensional signal space. If P, the number of basis vectors, is equal to N, the vector dimension, then the subspace defined by the input signal vectors 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 (6.27) 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 $\boldsymbol{y}^{\mathrm{T}}=[y(2), y(1), y(0), y(-1)]$ is the input signal, $\boldsymbol{x}^{\mathrm{T}}=[x(2), x(1), x(0)]$ is the desired signal and $\boldsymbol{w}^{\mathrm{T}}=[w_0, w_1]$ is the filter coefficient vector. As in Equation (6.26), the filter output can be written as

$$\begin{pmatrix} \hat{x}(2) \\ \hat{x}(1) \\ \hat{x}(0) \end{pmatrix} = w_0 \begin{pmatrix} y(2) \\ y(1) \\ y(0) \end{pmatrix} + w_1 \begin{pmatrix} y(1) \\ y(0) \\ y(-1) \end{pmatrix} \tag{6.28}$$

In Equation (6.28), the input signal vectors $\boldsymbol{y}_1^{\mathrm{T}}=[y(2),\ y(1),\ y(0)]$ and $\boldsymbol{y}_2^{\mathrm{T}}=[y(1),\ y(0),\ y(-1)]$ 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 signal 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 $\boldsymbol{y}_2$.

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 Equation (6.5) gives the least mean square error:

$$\begin{aligned} \mathcal{E}[e^2(m)] &= r_{xx}(0) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{r}_{yx} \\ &= r_{xx}(0) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w} \end{aligned} \tag{6.29}$$

Now, for zero-mean signals, it is easy to show that in Equation (6.29) the term $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w}$ is the variance of the Wiener filter output $\hat{x}(m)$:

$$\sigma_{\hat{x}}^2 = \mathcal{E}[\hat{x}^2(m)] = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{R}_{yy}\boldsymbol{w} \tag{6.30}$$

Therefore Equation (6.29) may be written as

$$\sigma_e^2 = \sigma_x^2 - \sigma_{\hat{x}}^2 \tag{6.31}$$

where $\sigma_x^2 = \mathcal{E}[x^2(m)], \sigma_{\hat{x}}^2 = \mathcal{E}[\hat{x}^2(m)]$ and $\sigma_e^2 = \mathcal{E}[e^2(m)]$ are the variances of the desired signal, the filter 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 $n(m)$:

$$y(m) = x_c(m) + n(m) \tag{6.32}$$

where the signal $x_c(m)$ is the part of the observation that is correlated with the desired signal $x(m)$, and it is this part of the input signal that may be transformable through a Wiener filter to the desired signal. Using Equation (6.32) the Wiener filter error may be decomposed into two distinct components:

$$\begin{aligned} e(m) &= x(m) - \sum_{k=0}^{P} w_k\, y(m-k) \\ &= \left[x(m) - \sum_{k=0}^{P} w_k\, x_c(m-k) \right] - \sum_{k=0}^{P} w_k\, n(m-k) \end{aligned} \tag{6.33}$$

or

$$e(m) = e_x(m) + e_n(m) \tag{6.34}$$

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.

$$e_x(m) = x(m) - \sum_{k=0}^{P-1} w_k\, x_c(m-k) \tag{6.35}$$

and $e_n(m)$ is the error in the output due to the presence of noise $n(m)$ in the input signal:

$$e_n(m) = -\sum_{k=0}^{P-1} w_k\, n(m-k) \tag{6.36}$$

The variance of filter error can be rewritten as

$$\sigma_e^2 = \sigma_{e_x}^2 + \sigma_{e_n}^2 \tag{6.37}$$

Note that in Equation (6.34), $e_x(m)$ is that part of the signal that cannot be recovered by the Wiener filter, and represents distortion in the signal output, 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:

(a) The spectra of the signal and the noise are separable by a linear filter.
(b) The signal component of the input, that is $x_c(m)$, is *linearly* transformable to $x(m)$.
(c) The filter length P is sufficiently large. The issue of signal and noise separability is addressed in Section 6.6.

6.5 Formulation of Wiener Filters in the Frequency Domain

In the frequency domain, the Wiener filter output $\hat{X}(f)$ is the product of the input signal $Y(f)$ and the filter frequency response $W(f)$:

$$\hat{X}(f)=W(f)Y(f) \tag{6.38}$$

The estimation error signal $E(f)$ is defined as the difference between the desired signal $X(f)$ and the filter output $\hat{X}(f)$,

$$\begin{aligned} E(f)&=X(f)-\hat{X}(f) \\ &=X(f)-W(f)Y(f) \end{aligned} \tag{6.39}$$

and the mean square error at a frequency f is given by

$$\mathcal{E}\left[\left|E(f)\right|^2\right]=\mathcal{E}\left[\left(X(f)-W(f)Y(f)\right)^*\left(X(f)-W(f)Y(f)\right)\right] \tag{6.40}$$

where $\mathcal{E}[\cdot]$ is the expectation function, and the symbol * denotes the complex conjugate. Note from Parseval's theorem that the mean square error in time and frequency domains are related by

$$\sum_{m=0}^{N-1} e^2(m) = \int_{-1/2}^{1/2} |E(f)|^2 \, df \tag{6.41}$$

To obtain the least mean square error filter we set the complex derivative of Equation (6.40) with respect to filter $W(f)$ to zero

$$\frac{\partial \mathcal{E}[|E(f)|^2]}{\partial W(f)} = 2W(f)P_{YY}(f) - 2P_{XY}(f) = 0 \tag{6.42}$$

where $P_{YY}(f)=\mathcal{E}[Y(f)Y^*(f)]$ and $P_{XY}(f)=\mathcal{E}[X(f)Y^*(f)]$ are the power spectrum of $Y(f)$, and the cross-power spectrum of $Y(f)$ and $X(f)$ respectively. From Equation (6.42), the least mean square error Wiener filter in the frequency domain is given as

$$W(f) = \frac{P_{XY}(f)}{P_{YY}(f)} \tag{6.43}$$

Alternatively, the frequency-domain Wiener filter Equation (6.43) can be obtained from the Fourier transform of the time-domain Wiener Equation (6.9):

$$\sum_m \sum_{k=0}^{P-1} w_k r_{yy}(m-k) e^{-j\omega m} = \sum_m r_{yx}(n) e^{-j\omega m} \tag{6.44}$$

From the Wiener–Khinchine relation, the 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, it is easy to show that the Wiener filter is given by Equation (6.43).

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 multisensor systems, and in channel equalisation.

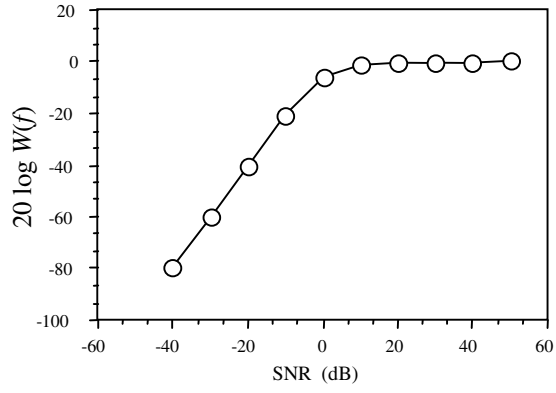


Figure 6.4 Variation of the gain of Wiener filter frequency response with SNR.

6.6.1 Wiener Filter for Additive Noise Reduction

Consider a signal $x(m)$ observed in a broadband additive noise $n(m)$., and model as

$$y(m) = x(m) + n(m) \tag{6.45}$$

Assuming that the signal and the noise are uncorrelated, 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 $n(m)$:

$$\boldsymbol{R}_{yy} = \boldsymbol{R}_{xx} + \boldsymbol{R}_{nn} \tag{6.46}$$

and we can also write

$$\boldsymbol{r}_{xy} = \boldsymbol{r}_{xx} \tag{6.47}$$

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 (6.46) and (6.47) in the Wiener filter, Equation (6.10), yields

$$\boldsymbol{w} = (\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})^{-1} \boldsymbol{r}_{xx} \tag{6.48}$$

Equation (6.48) 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

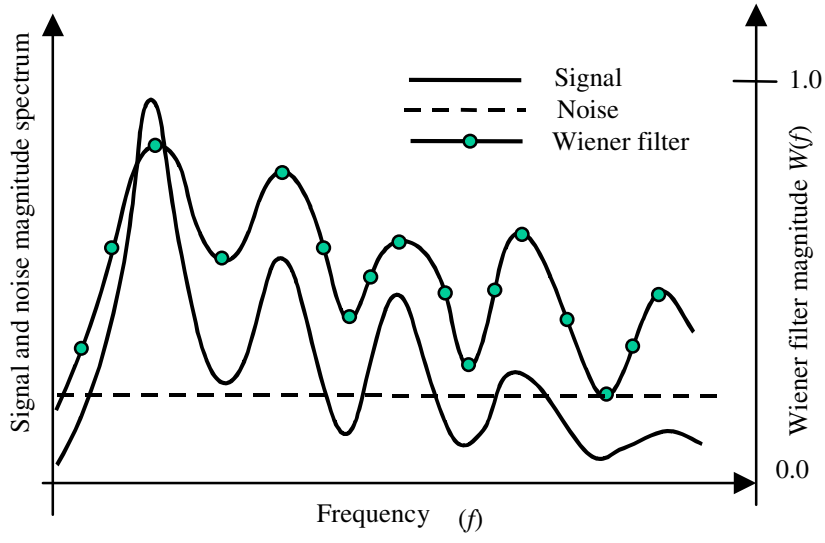


Figure 6.5 Illustration of the variation of Wiener frequency response with signal spectrum for additive white noise. The Wiener filter response broadly follows the signal spectrum.

$$Y(f)=X(f)+N(f) \tag{6.49}$$

where $X(f)$ and $N(f)$ are the signal and noise spectra. For a signal observed in additive random noise, the frequency-domain Wiener filter is obtained as

$$W(f)=\frac{P_{XX}(f)}{P_{XX}(f)+P_{NN}(f)} \tag{6.50}$$

where $P_{XX}(f)$ and $P_{NN}(f)$ are the signal and noise power spectra. Dividing the numerator and the denominator of Equation (6.50) by the noise power spectra $P_{NN}(f)$ and substituting the variable $SNR(f)=P_{XX}(f)/P_{NN}(f)$ yields

$$W(f)=\frac{SNR(f)}{SNR(f)+1} \tag{6.51}$$

where SNR is a signal-to-noise ratio measure. Note that the variable, $SNR(f)$ is expressed in terms of the power-spectral ratio, and not in the more usual terms of log power ratio. Therefore $SNR(f)=0$ corresponds to $-\infty$ *dB*.

From Equation (6.51), the following interpretation of the Wiener filter frequency response $W(f)$ in terms of the signal-to-noise ratio can be

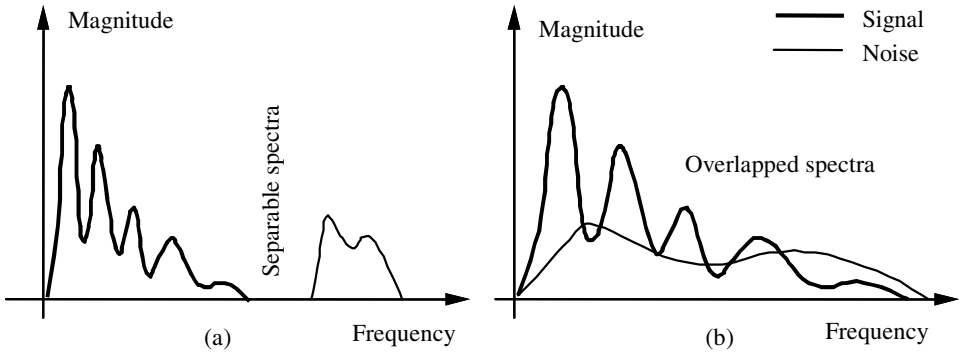


Figure 6.6 Illustration of separability: (a) The signal and noise spectra do not overlap, and the signal can be recovered by a low-pass filter; (b) the signal and noise spectra overlap, and the noise can be reduced but not completely removed.

deduced. For additive noise, the Wiener filter frequency response is a real positive number in the range $0 \leq W(f) \leq 1$. Now consider the two limiting cases of (a) a noise-free signal $SNR(f) = \infty$ 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*. Figure 6.4 shows the variation of the Wiener filter response $W(f)$, with the signal-to-noise ratio $SNR(f)$.

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.

6.6.2 Wiener Filter and the Separability of Signal and Noise

A signal is completely recoverable from noise if the spectra of the signal and the noise do not overlap. An example of a noisy signal with separable signal and noise spectra is shown in Figure 6.6(a). In this case, the signal

and the noise occupy different parts of the frequency spectrum, and can be separated with a low-pass, or a high-pass, filter. Figure 6.6(b) illustrates a more common example of a signal and noise process with overlapping spectra. For this case, it is not possible to completely separate the signal from the noise. However, 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 (6.51).

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 $X(f)$ and the filter response $W(f)$ as expressed in Equation (6.38). Taking the expectation of the squared magnitude of both sides of Equation (6.38) yields the power spectrum of the filtered signal as

$$\begin{aligned} \mathcal{E}[|\hat{X}(f)|^2] &= |W(f)|^2 \mathcal{E}[|Y(f)|^2] \\ &= |W(f)|^2 P_{YY}(f) \end{aligned} \tag{6.52}$$

Substitution of $W(f)$ from Equation (6.43) in Equation (6.52) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XY}^2(f)}{P_{YY}(f)} \tag{6.53}$$

Now, for a signal observed in an uncorrelated additive noise we have

$$P_{YY}(f) = P_{XX}(f) + P_{NN}(f) \tag{6.54}$$

and

$$P_{XY}(f) = P_{XX}(f) \tag{6.55}$$

Substitution of Equations (6.54) and (6.55) in Equation (6.53) yields

$$\mathcal{E}[|\hat{X}(f)|^2] = \frac{P_{XX}^2(f)}{P_{XX}(f) + P_{NN}(f)} \tag{6.56}$$

Now, in Equation (6.38) if instead of the Wiener filter, the square root of the Wiener filter magnitude frequency response is used, the result is

$$\hat{X}(f)=|W(f)|^{1/2}Y(f) \tag{6.57}$$

and the power spectrum of the signal, filtered by the square-root Wiener filter, is given by

$$\mathcal{E}[|\hat{X}(f)|^2]=\left[|W(f)|^{1/2}\right]^2\mathcal{E}[|Y(f)|^2]=\frac{P_{XY}(f)}{P_{YY}(f)}P_{YY}(f)=P_{XY}(f) \tag{6.58}$$

Now, for uncorrelated signal and noise Equation (6.58) becomes

$$\mathcal{E}[|\hat{X}(f)|^2]=P_{XX}(f) \tag{6.59}$$

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

Communication channel distortions 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

$$y(m)=\sum_{k=0}^{P-1}h_k\,x(m-k)+n(m) \tag{6.60}$$

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 (6.60) becomes

$$Y(f)=X(f)H(f)+N(f) \tag{6.61}$$

where $X(f)$, $Y(f)$, $H(f)$ and $N(f)$ are the signal, noisy signal, channel and noise spectra respectively. To remove the channel distortions, the receiver is followed by an equaliser. The equaliser input is the distorted channel output, and the desired signal is the channel input. Using Equation (6.43) it is easy to show that the Wiener equaliser in the frequency domain is given by

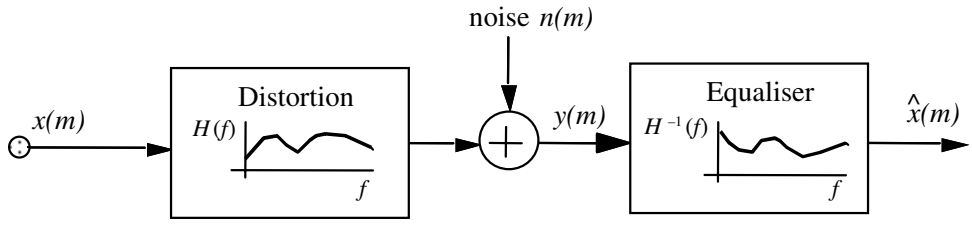


Figure 6.7 Illustration of a channel model followed by an equaliser.

$$W(f)=\frac{P_{XX}(f)H^{*}(f)}{P_{XX}(f)|H(f)|^{2}+P_{NN}(f)} \tag{6.62}$$

where it is assumed that the channel noise and the signal are uncorrelated. In the absence of channel noise, $P_{NN}(f)=0$, and the Wiener filter is simply the inverse of the channel filter model $W(f)=H^{-1}(f)$. The equalisation problem is treated in detail in Chapter 15.

6.6.5 Time-Alignment of Signals in Multichannel/Multisensor Systems

In multichannel/multisensor signal processing there are a number 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. As a simple example, consider the problem of time-alignment of two noisy records of a signal given as

$$y_1(m)=x(m)+n_1(m) \tag{6.63}$$

$$y_2(m)=Ax(m-D)+n_2(m) \tag{6.64}$$

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 time delay of arrival of the two signals, and A is an amplitude scaling factor. Now assume that $y_1(m)$ is used as the input 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

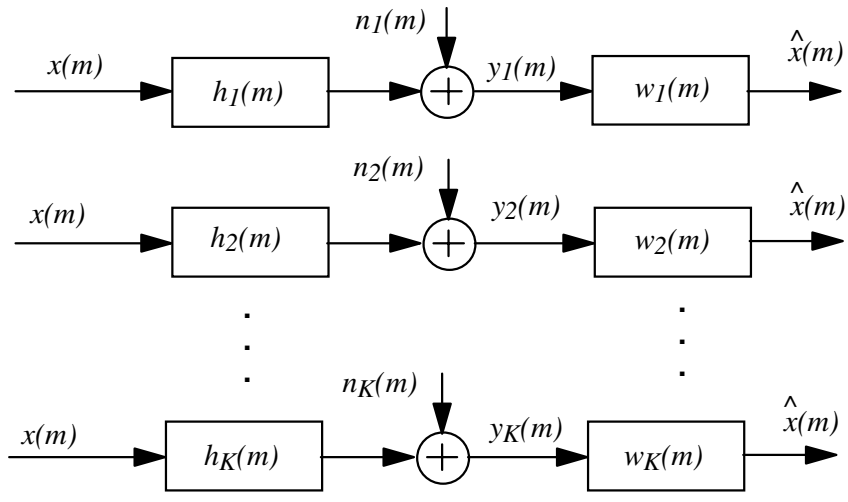


Figure 6.8 Illustration of a multichannel system where Wiener filters are used to time-align the signals from different channels.

$$\begin{aligned} e(m) &= y_2(m) - \sum_{k=0}^{P-1} w_k\, y_1(m) \\ &= \left(Ax(m-D) - \sum_{k=0}^{P-1} w_k\, x(m) \right) + \left(\sum_{k=0}^{P-1} w_k\, n_1(m) \right) + n_2(m) \end{aligned} \tag{6.65}$$

The Wiener filter strives to minimise the terms shown inside the square brackets in Equation (6.65). Using the Wiener filter Equation (6.10), we have

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{R}_{y_1 y_1}^{-1} \boldsymbol{r}_{y_1 y_2} \\ &= \left(\boldsymbol{R}_{xx} + \boldsymbol{R}_{n_1 n_1} \right)^{-1} A \boldsymbol{r}_{xx}(D) \end{aligned} \tag{6.66}$$

where $\boldsymbol{r}_{xx}(D)=\mathcal{E}\,[x(PD)\boldsymbol{x}(m)]$. The frequency-domain equivalent of Equation (6.65) can be derived as

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{N_1 N_1}(f)} A e^{-j\omega D} \tag{6.67}$$

Note that in the absence of noise, the Wiener filter becomes a pure phase (or a pure delay) filter with a flat magnitude response.

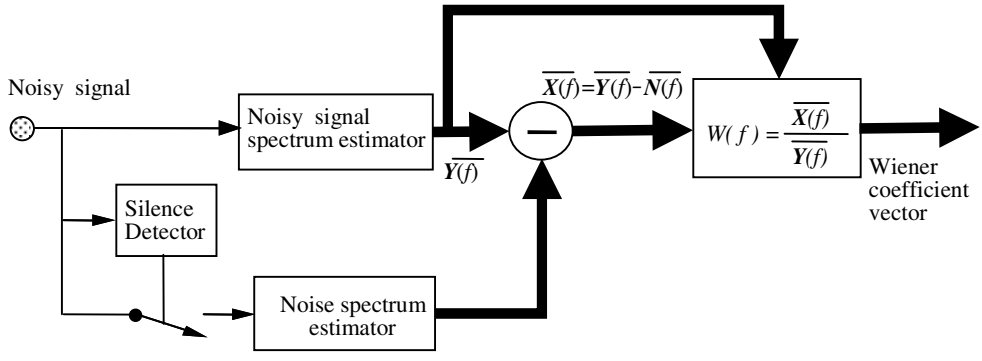


Figure 6.9 Configuration of a system for estimation of frequency Wiener filter.

6.6.6 Implementation of Wiener Filters

The implementation of a Wiener filter for additive noise reduction, using Equations (6.48)–(6.50), requires the autocorrelation functions, or equivalently the power spectra, of the signal and noise. 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 remains 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, 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. An estimate of the desired signal power spectra is 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 bands of frequencies. 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.

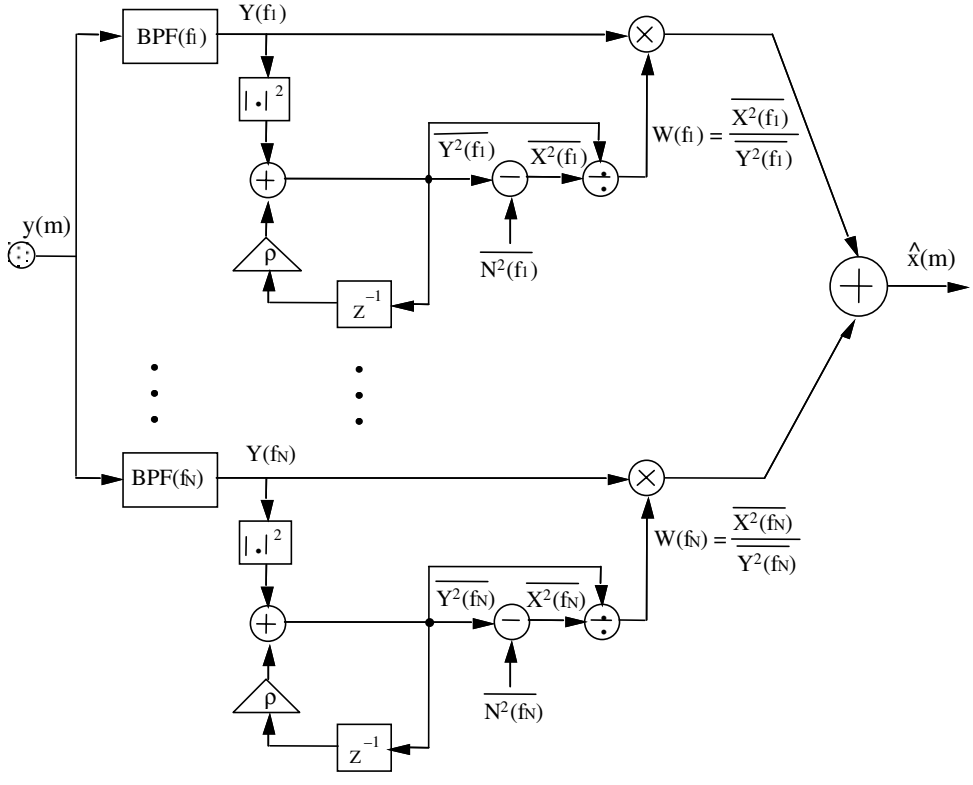


Figure 6.10 A filter-bank implementation of a Wiener filter.

6.7 The Choice of Wiener Filter Order

The choice of Wiener filter order affects:

(a) the ability of the filter to remove distortions and reduce the noise;
(b) the computational complexity of the filter; and
(c) the numerical stability of the of the Wiener solution, 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, the number of filter coefficients is equal to the number of filter banks, and typically the

number of filter banks is between 16 to 64. On the other hand for many applications, a direct implementation of the time-domain Wiener filter requires a larger filter length say between 64 and 256 taps.

A reduction in the required length of a time-domain Wiener filter can be achieved by dividing the time domain signal into N sub-band signals. Each sub-band signal can then be decimated by a factor of N. The decimation results in a reduction, by a factor of N, in the required length of each sub-band Wiener filter. In Chapter 14, a subband echo canceller is described.

6.8 Summary

A Wiener filter is formulated to map an input signal to an output that is as close to a desired signal as possible. This chapter began with the derivation of the least square error Wiener filter. In Section 6.2, we derived the block-data least square error Wiener filter for applications where only finite-length realisations of the input and the desired signals are available. In such cases, the filter is obtained by minimising a time-averaged squared error function. In Section 6.3, we considered a vector space interpretation of the Wiener filters as the perpendicular projection of the desired signal onto the space of the input signal.

In Section 6.4, the least mean square error signal was analysed. The mean square error is zero only if the input signal is related to the desired signal through a linear and invertible filter. For most cases, owing to noise and/or nonlinear distortions of the input signal, the minimum mean square error would be 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, pp. 716–23.

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. Wiley, New York.

DURBIN J. (1959) Efficient Estimation of Parameters in Moving Average Models. Biometrica, **46**, pp. 306–16.

GIORDANO A.A. and HSU F.M. (1985) Least Square Estimation with Applications to Digital Signal Processing. Wiley, New York.

GIVENS W. (1958) Computation of Plane Unitary Rotations Transforming a General Matrix to Triangular Form. SIAM J. Appl. Math. **6**, pp. 26–50.

GOLUB G.H. and REINSCH (1970) Singular Value Decomposition and Least Squares Solutions. Numerical Mathematics, **14**, pp. 403–20.

GOLUB G.H. and VAN LOAN C.F. (1983). Matrix Computations. Johns Hopkins University Press, Baltimore, MD.

GOLUB G.H., and VAN LOAN C.F. (1980). An Analysis of the Total Least Squares Problem. SIAM Journal of Numerical Analysis, **17**, pp. 883–93.

HALMOS P.R. (1974) Finite-Dimensional Vector Spaces. Springer-Verlag, New York.

HAYKIN S. (1991) Adaptive Filter Theory, 2nd Ed. 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**, pp. 146–81.

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**, pp. 164-76.

KOLMOGROV A.N. (1939) Sur l' Interpolation et Extrapolation des Suites Stationaires. Comptes Rendus de l'Academie des Sciences, **208**, pp. 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 Procesing: An Introduction, 2nd Ed. 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 Ed. Harcourt Brace Jovanovich, San Diego, California.

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.

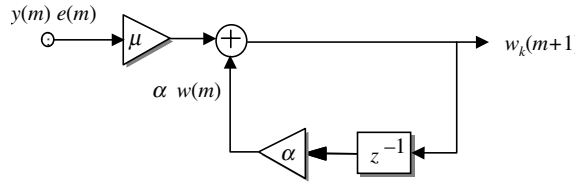
WHITTLE P.W. (1983) Prediction and Regulation by Linear Least-Squares Methods. University of Minnesota Press, Minneapolis, Minnesota.

WOLD H. (1954) The Analysis of Stationary Time Series, 2nd Ed. Almquist and Wicksell, Uppsala.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

7



ADAPTIVE FILTERS

Adaptive filters are used for non-stationary signals and environments, or in applications where a sample-by-sample adaptation of a process or a low processing delay is required. Applications of adaptive filters include multichannel noise reduction, radar/sonar signal processing, channel equalization for cellular mobile phones, echo cancellation, and low delay speech coding. This chapter begins with a study of the state-space Kalman filter. In Kalman theory a state equation models the dynamics of the signal generation process, and an observation equation models the channel distortion and additive noise. Then we consider recursive least square (RLS) error adaptive filters. The RLS filter is a sample-adaptive formulation of the Wiener filter, and for stationary signals should converge to the same solution as the Wiener filter. In least square error filtering, an alternative to using a Wiener-type closed-form solution is an iterative gradient-based search for the optimal filter coefficients. The steepest-descent search is a gradient-based method for searching the least square error performance curve for the minimum error filter coefficients. We study the steepest-descent method, and then consider the computationally inexpensive LMS gradient search method.

7.1 State-Space Kalman Filters

The Kalman filter is a recursive least square error method for estimation of a signal distorted in transmission through a channel and observed in noise. Kalman filters can be used with time-varying as well as time-invariant processes. Kalman filter theory is based on a state-space approach in which a state equation models the dynamics of the signal process and an observation equation models the noisy observation signal. For a signal $\boldsymbol{x}(m)$ and noisy observation $\boldsymbol{y}(m)$, the state equation model and the observation model are defined as

$$\boldsymbol{x}(m) = \boldsymbol{\Phi}(m, m-1)\boldsymbol{x}(m-1) + \boldsymbol{e}(m) \tag{7.1}$$

$$\boldsymbol{y}(m) = \boldsymbol{H}(m)\boldsymbol{x}(m) + \boldsymbol{n}(m) \tag{7.2}$$

where

$\boldsymbol{x}(m)$ is the P-dimensional signal, or the state parameter, vector at time m,
$\boldsymbol{\Phi}(m, m-1)$ is a $P \times P$ dimensional state transition matrix that relates the states of the process at times m–1 and m,
$\boldsymbol{e}(m)$ is the P-dimensional uncorrelated input excitation vector of the state equation,
$\boldsymbol{\Sigma}_{\boldsymbol{ee}}(m)$ is the $P \times P$ covariance matrix of $\boldsymbol{e}(m)$,
$\boldsymbol{y}(m)$ is the M-dimensional noisy and distorted observation vector,
$\boldsymbol{H}(m)$ is the $M \times P$ channel distortion matrix,
$\boldsymbol{n}(m)$ is the M-dimensional additive noise process,
$\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)$ is the $M \times M$ covariance matrix of $\boldsymbol{n}(m)$.

The Kalman filter can be derived as a recursive minimum mean square error predictor of a signal $\boldsymbol{x}(m)$, given an observation signal $\boldsymbol{y}(m)$. The filter derivation assumes that the state transition matrix $\boldsymbol{\Phi}(m, m-1)$, the channel distortion matrix $\boldsymbol{H}(m)$, the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{ee}}(m)$ of the state equation input and the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)$ of the additive noise are given.

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. Now assume that $\hat{\boldsymbol{y}}(m|m-1)$ is the least square error prediction of $\boldsymbol{y}(m)$ based on the observations $[\boldsymbol{y}(0), \ldots, \boldsymbol{y}(m-1)]$. Define a so-called *innovation*, or prediction error signal as

$$\boldsymbol{v}(m) = \boldsymbol{y}(m) - \hat{\boldsymbol{y}}(m|m-1) \tag{7.3}$$

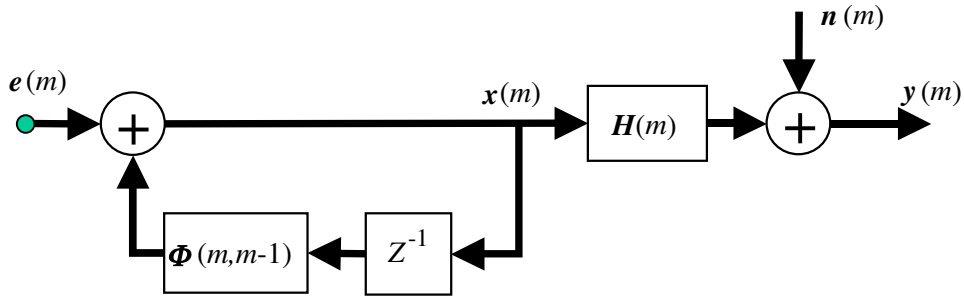


Figure 7.1 Illustration of signal and observation models in Kalman filter theory.

The innovation signal vector $\boldsymbol{v}(m)$ contains all that is unpredictable from the past observations, including both the noise and the unpredictable part of the signal. For an optimal linear least mean square error estimate, the innovation signal must be uncorrelated and orthogonal to the past observation vectors; hence we have

$$\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{y}^{\mathrm{T}}(m-k)\right]=0, \qquad k>0 \tag{7.4}$$

and

$$\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(k)\right]=0, \qquad m\neq k \tag{7.5}$$

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 (7.4) is used as the starting point to derive an optimal linear filter whose innovations are orthogonal to the past observations.*

Substituting the observation Equation (7.2) in Equation (7.3) and using the relation

$$\begin{aligned}\hat{\boldsymbol{y}}(m\,|\,m-1)&=\mathcal{E}\left[\boldsymbol{y}(m)\,\middle|\,\hat{\boldsymbol{x}}(m|m-1)\right]\\&=\boldsymbol{H}(m)\hat{\boldsymbol{x}}(m|m-1)\end{aligned} \tag{7.6}$$

yields

$$\begin{aligned}\boldsymbol{v}(m)&=\boldsymbol{H}(m)\boldsymbol{x}(m)+\boldsymbol{n}(m)-\boldsymbol{H}(m)\,\hat{\boldsymbol{x}}(m|m-1)\\&=\boldsymbol{H}(m)\tilde{\boldsymbol{x}}(m)+\boldsymbol{n}(m)\end{aligned} \tag{7.7}$$

where $\tilde{\boldsymbol{x}}(m)$ is the signal prediction error vector defined as

$$\tilde{\boldsymbol{x}}(m)=\boldsymbol{x}(m)-\hat{\boldsymbol{x}}(m|m-1) \tag{7.8}$$

From Equation (7.7) the covariance matrix of the innovation signal is given by

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{vv}}(m) &= \mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(m)\right] \\ &= \boldsymbol{H}(m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{H}^{\mathrm{T}}(m) + \boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)\end{aligned} \tag{7.9}$$

where $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)$ is the covariance matrix of the prediction error $\tilde{\boldsymbol{x}}(m)$. Let $\hat{\boldsymbol{x}}(m+1|m)$ denote the least square error prediction of the signal $\boldsymbol{x}(m+1)$. Now, the prediction of $\boldsymbol{x}(m+1)$, based on the samples available up to the time m, can be expressed recursively as a linear combination of the prediction based on the samples available up to the time $m-1$ and the innovation signal at time m as

$$\hat{\boldsymbol{x}}(m+1|m) = \hat{\boldsymbol{x}}(m+1|m-1) + \boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.10}$$

where the $P \times M$ matrix $\boldsymbol{K}(m)$ is the Kalman gain matrix. Now, from Equation (7.1), we have

$$\hat{\boldsymbol{x}}(m+1|m-1) = \boldsymbol{\Phi}(m+1,m)\hat{\boldsymbol{x}}(m|m-1) \tag{7.11}$$

Substituting Equation (7.11) in (7.10) gives a recursive prediction equation as

$$\hat{\boldsymbol{x}}(m+1|m) = \boldsymbol{\Phi}(m+1,m)\hat{\boldsymbol{x}}(m|m-1) + \boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.12}$$

To obtain a recursive relation for the computation and update of the Kalman gain matrix, we multiply both sides of Equation (7.12) by $\boldsymbol{v}^{\mathrm{T}}(m)$ and take the expectation of the results to yield

$$\mathcal{E}\left[\hat{\boldsymbol{x}}(m+1|m)\boldsymbol{v}^{\mathrm{T}}(m)\right] = \mathcal{E}\left[\boldsymbol{\Phi}(m+1,m)\hat{\boldsymbol{x}}(m|m-1)\boldsymbol{v}^{\mathrm{T}}(m)\right] + \boldsymbol{K}(m)\mathcal{E}\left[\boldsymbol{v}(m)\boldsymbol{v}^{\mathrm{T}}(m)\right] \tag{7.13}$$

Owing to the required orthogonality of the innovation sequence and the past samples, we have

$$\mathcal{E}\left[\hat{\boldsymbol{x}}(m|m-1)\boldsymbol{v}^{\mathrm{T}}(m)\right] = 0 \tag{7.14}$$

Hence, from Equations (7.13) and (7.14), the Kalman gain matrix is given by

$$\boldsymbol{K}(m) = \mathcal{E}\left[\hat{\boldsymbol{x}}(m+1|m)\boldsymbol{v}^{\mathrm{T}}(m)\right]\boldsymbol{\Sigma}_{\boldsymbol{vv}}^{-1}(m) \tag{7.15}$$

The first term on the right-hand side of Equation (7.15) can be expressed as

$$\begin{aligned}\mathcal{E}\left[\hat{\boldsymbol{x}}(m+1|m)\boldsymbol{v}^{\mathrm{T}}(m)\right] &= \mathcal{E}\left[\left(\boldsymbol{x}(m+1)-\tilde{\boldsymbol{x}}(m+1|m)\right)\boldsymbol{v}^{\mathrm{T}}(m)\right] \\ &= \mathcal{E}\left[\boldsymbol{x}(m+1)\boldsymbol{v}^{\mathrm{T}}(m)\right] \\ &= \mathcal{E}\left[\left(\boldsymbol{\Phi}(m+1,m)\boldsymbol{x}(m)+\boldsymbol{e}(m+1)\right)\left(\boldsymbol{y}(m)-\hat{\boldsymbol{y}}(m|m-1)\right)^{\mathrm{T}}\right] \\ &= \mathcal{E}\left[\left[\boldsymbol{\Phi}(m+1,m)\left(\hat{\boldsymbol{x}}(m|m-1)+\tilde{\boldsymbol{x}}(m|m-1)\right)\right]\left(\boldsymbol{H}(m)\tilde{\boldsymbol{x}}(m|m-1)+\boldsymbol{n}(m)\right)^{\mathrm{T}}\right] \\ &= \boldsymbol{\Phi}(m+1,m)\mathcal{E}\left[\tilde{\boldsymbol{x}}(m|m-1)\tilde{\boldsymbol{x}}^{\mathrm{T}}(m|m-1)\right]\boldsymbol{H}^{\mathrm{T}}(m)\end{aligned} \tag{7.16}$$

In developing the successive lines of Equation (7.16), we have used the following relations:

$$\mathcal{E}\left[\tilde{\boldsymbol{x}}(m+1|m)\boldsymbol{v}^{\mathrm{T}}(m)\right]=0 \tag{7.17}$$

$$\mathcal{E}\left[\boldsymbol{e}(m+1)\left(\boldsymbol{y}(m)-\hat{\boldsymbol{y}}(m|m-1)\right)^{\mathrm{T}}\right]=0 \tag{7.18}$$

$$\boldsymbol{x}(m)=\hat{\boldsymbol{x}}(m|m-1)+\tilde{\boldsymbol{x}}(m|m-1) \tag{7.19}$$

$$\mathcal{E}\left[\hat{\boldsymbol{x}}(m|m-1)\tilde{\boldsymbol{x}}(m|m-1)\right]=0 \tag{7.20}$$

and we have also used the assumption that the signal and the noise are uncorrelated. Substitution of Equations (7.9) and (7.16) in Equation (7.15) yields the following equation for the Kalman gain matrix:

$$\boldsymbol{K}(m)=\boldsymbol{\Phi}(m+1,m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{H}^{\mathrm{T}}(m)\left[\boldsymbol{H}(m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{H}^{\mathrm{T}}(m)+\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)\right]^{-1} \tag{7.21}$$

where $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)$ is the covariance matrix of the signal prediction error $\tilde{\boldsymbol{x}}(m|m-1)$. To derive a recursive relation for $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)$, we consider

$$\tilde{\boldsymbol{x}}(m|m-1)=\boldsymbol{x}(m)-\hat{\boldsymbol{x}}(m|m-1) \tag{7.22}$$

Substitution of Equation (7.1) and (7.12) in Equation (7.22) and rearrangement of the terms yields

$$\begin{aligned}\tilde{\boldsymbol{x}}(m|m-1) &= \left[\boldsymbol{\Phi}(m,m-1)\boldsymbol{x}(m-1)+\boldsymbol{e}(m)\right]-\left[\boldsymbol{\Phi}(m,m-1)\hat{\boldsymbol{x}}(m-1|m-2)+\boldsymbol{K}(m-1)\boldsymbol{v}(m-1)\right] \\ &= \boldsymbol{\Phi}(m,m-1)\tilde{\boldsymbol{x}}(m-1)+\boldsymbol{e}(m)-\boldsymbol{K}(m-1)\boldsymbol{H}(m-1)\tilde{\boldsymbol{x}}(m-1)+\boldsymbol{K}(m-1)\boldsymbol{n}(m-1) \\ &= \left[\boldsymbol{\Phi}(m,m-1)-\boldsymbol{K}(m-1)\boldsymbol{H}(m-1)\right]\tilde{\boldsymbol{x}}(m-1)+\boldsymbol{e}(m)+\boldsymbol{K}(m-1)\boldsymbol{n}(m-1)\end{aligned} \tag{7.23}$$

From Equation (7.23) we can derive the following recursive relation for the variance of the signal prediction error

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)=\boldsymbol{L}(m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m-1)\boldsymbol{L}^{\mathrm{T}}(m)+\boldsymbol{\Sigma}_{\boldsymbol{ee}}(m)+\boldsymbol{K}(m-1)\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m-1)\boldsymbol{K}^{\mathrm{T}}(m-1) \tag{7.24}$$

where the $P\times P$ matrix $\boldsymbol{L}(m)$ is defined as

$$\boldsymbol{L}(m)=\left[\boldsymbol{\Phi}(m,m-1)-\boldsymbol{K}(m-1)\boldsymbol{H}(m-1)\right] \tag{7.25}$$

Kalman Filtering Algorithm

Input: observation vectors $\{\boldsymbol{y}(m)\}$
Output: state or signal vectors $\{\hat{\boldsymbol{x}}(m)\}$
Initial conditions:

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(0)=\delta\mathbf{I} \tag{7.26}$$

$$\hat{\boldsymbol{x}}(0|-1)=0 \tag{7.27}$$

For $m = 0, 1, \ldots$
Innovation signal:

$$\boldsymbol{v}(m)=\boldsymbol{y}(m)-\boldsymbol{H}(m)\hat{\boldsymbol{x}}(m|m-1) \tag{7.28}$$

Kalman gain:

$$\boldsymbol{K}(m)=\boldsymbol{\Phi}(m+1,m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{H}^{\mathrm{T}}(m)\left[\boldsymbol{H}(m)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{H}^{\mathrm{T}}(m)+\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)\right]^{-1} \tag{7.29}$$

Prediction update:

$$\hat{\boldsymbol{x}}(m+1|m)=\boldsymbol{\Phi}(m+1,m)\hat{\boldsymbol{x}}(m|m-1)+\boldsymbol{K}(m)\boldsymbol{v}(m) \tag{7.30}$$

Prediction error correlation matrix update:

$$\boldsymbol{L}(m+1)=\left[\boldsymbol{\Phi}(m+1,m)-\boldsymbol{K}(m)\boldsymbol{H}(m)\right] \tag{7.31}$$

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m+1)=\boldsymbol{L}(m+1)\boldsymbol{\Sigma}_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}(m)\boldsymbol{L}(m+1)^{\mathrm{T}}+\boldsymbol{\Sigma}_{\boldsymbol{ee}}(m+1)+\boldsymbol{K}(m)\boldsymbol{\Sigma}_{\boldsymbol{nn}}(m)\boldsymbol{K}(m) \tag{7.32}$$

Example 7.1 Consider the Kalman filtering of a first-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

$$x(m)=a(m)x(m-1)+e(m) \tag{7.33}$$

$$y(m)=x(m)+n(m) \tag{7.34}$$

Let $\sigma_e^2(m)$ and $\sigma_n^2(m)$ denote the variances of the excitation signal $e(m)$ and the noise $n(m)$ respectively. Substituting $\Phi(m+1,m)=a(m)$ and $H(m)=1$ in the Kalman filter equations yields the following Kalman filter algorithm:

Initial conditions:

$$\sigma_{\tilde{x}}^2(0) = \delta \tag{7.35}$$

$$\hat{x}(0|-1)= 0 \tag{7.36}$$

For $m = 0, 1, ...$
Kalman gain:

$$k(m)=\frac{a(m+1)\sigma_{\tilde{x}}^2(m)}{\sigma_{\tilde{x}}^2(m)+\sigma_n^2(m)} \tag{7.37}$$

Innovation signal:

$$v(m)= y(m)-\hat{x}(m\,|\,m-1) \tag{7.38}$$

Prediction signal update:

$$\hat{x}(m+1|m)= a(m+1)\hat{x}(m|m-1)+k(m)v(m) \tag{7.39}$$

Prediction error update:

$$\sigma_{\tilde{x}}^2(m+1) = [a(m+1)-k(m)]^2\,\sigma_{\tilde{x}}^2(m)+\sigma_e^2(m+1)+k^2(m)\,\sigma_n^2(m) \tag{7.40}$$

where $\sigma_{\tilde{x}}^2(m)$ is the variance of the prediction error signal.

Example 7.2 *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

$$x(m)=x(m-1)=x \tag{7.41}$$

$$y(m)=x+n(m) \tag{7.42}$$

Note that $\Phi(m,m-1)=1$, state excitation $e(m)=0$ and $H(m)=1$. Using the Kalman algorithm, we have the following recursive solutions:

Initial Conditions:

$$\sigma_{\tilde{x}}^2(0) = \delta \tag{7.43}$$

$$\hat{x}(0|-1) = 0 \tag{7.44}$$

For $m = 0, 1, ...$
Kalman gain:

$$k(m)=\frac{\sigma_{\tilde{x}}^2(m)}{\sigma_{\tilde{x}}^2(m)+\sigma_n^2(m)} \tag{7.45}$$

Innovation signal:

$$v(m)= y(m)-\hat{x}(m| m-1) \tag{7.46}$$

Prediction signal update:

$$\hat{x}(m+1| m)=\hat{x}(m | m-1)+ k(m)v(m) \tag{7.47}$$

Prediction error update:

$$\sigma_{\tilde{x}}^2(m+1)=[1-k(m)]^2\sigma_{\tilde{x}}^2(m)+ k^2(m)\sigma_n^2(m) \tag{7.48}$$

7.2 Sample-Adaptive Filters

Sample 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 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 environments are important objectives.

Figure 7.2 illustrates the configuration of a least square error adaptive filter. At each sampling time, an adaptation algorithm adjusts the filter coefficients to minimise the difference between the filter output and a desired, or target, signal. An adaptive filter starts at some initial state, and 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:

next parameter estimate = previous parameter estimate + update(*error*)

where the update term is a function of the error signal. In adaptive filtering a number of decisions has to be made concerning the filter model and the adaptation algorithm:

(a) 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 this reason are the type most often used in practice.

(b) 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.

(c) Adaptation algorithm: The two most widely 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 condition, the minimum error at convergence, the numerical stability and the robustness of the algorithm to initial parameter states.

7.3 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 equalization, echo cancellation and radar where the filter should be able to track relatively fast changes in the signal process.

In the recursive least square algorithm, the adaptation starts with some initial filter state, and successive samples of the input signals are used to adapt the filter coefficients. Figure 7.2 illustrates the configuration of an adaptive filter where $y(m)$, $x(m)$ and $\boldsymbol{w}(m)=[w_0(m), w_1(m), ..., w_{P-1}(m)]$ denote the filter input, the desired signal and the filter coefficient vector respectively. The filter output can be expressed as

$$\hat{x}(m) = \boldsymbol{w}^{\mathrm{T}}(m)\,\boldsymbol{y}(m) \tag{7.49}$$

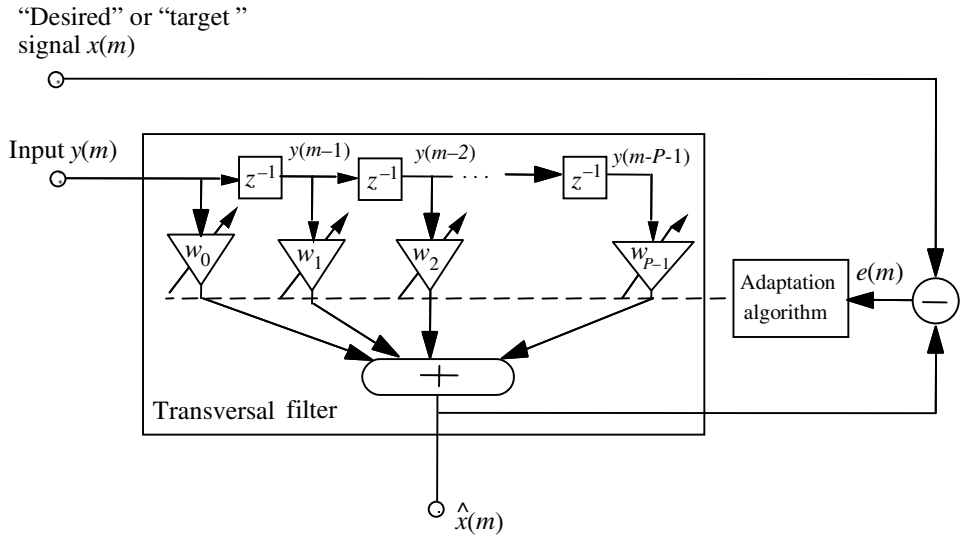


Figure 7.2 Illustration of the configuration of an adaptive filter.

where $\hat{x}(m)$ is an estimate of the desired signal $x(m)$. The filter error signal is defined as

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m) \end{aligned} \tag{7.50}$$

The adaptation process is based on the minimization of the mean square error criterion defined as

$$\begin{aligned} \mathcal{E}[e^2(m)] &= \mathcal{E}\left\{\left[x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)\right]^2\right\} \\ &= \mathcal{E}[x^2(m)] - 2\boldsymbol{w}^{\mathrm{T}}(m)\mathcal{E}[\boldsymbol{y}(m)x(m)] + \boldsymbol{w}^{\mathrm{T}}(m)\mathcal{E}[\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)]\,\boldsymbol{w}(m) \\ &= r_{xx}(0) - 2\boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{r}_{\boldsymbol{yx}}(m) + \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{R}_{\boldsymbol{yy}}(m)\boldsymbol{w}(m) \end{aligned} \tag{7.51}$$

The Wiener filter is obtained by minimising the mean square error with respect to the filter coefficients. For stationary signals, the result of this minimisation is given in Chapter 6, Equation (6.10), as

$$\boldsymbol{w} = \boldsymbol{R}_{\boldsymbol{yy}}^{-1}\,\boldsymbol{r}_{\boldsymbol{yx}} \tag{7.52}$$

where $\boldsymbol{R}_{yy}$ is the autocorrelation matrix of the input signal and $\boldsymbol{r}_{yx}$ is the cross-correlation vector of the input and the target signals. In the following, we formulate a recursive, time-update, adaptive formulation of Equation (7.52). From Section 6.2, for a block of N sample vectors, the correlation matrix can be written as

$$\boldsymbol{R}_{yy} = \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} = \sum_{m=0}^{N-1} \boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.53}$$

where $\boldsymbol{y}(m)=[y(m), ..., y(m-P)]^{\mathrm{T}}$. Now, the sum of vector product in Equation (7.53) can be expressed in recursive fashion as

$$\boldsymbol{R}_{yy}(m) = \boldsymbol{R}_{yy}(m-1) + \boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.54}$$

To introduce adaptability to the time variations of the signal statistics, the autocorrelation estimate in Equation (7.54) can be windowed by an exponentially decaying window:

$$\boldsymbol{R}_{yy}(m) = \lambda\,\boldsymbol{R}_{yy}(m-1) + \boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m) \tag{7.55}$$

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

$$\boldsymbol{r}_{yx} = \sum_{m=0}^{N-1} \boldsymbol{y}(m)x(m) \tag{7.56}$$

The sum of products in Equation (7.56) can be calculated in recursive form as

$$\boldsymbol{r}_{yx}(m) = \boldsymbol{r}_{yx}(m-1) + \boldsymbol{y}(m)x(m) \tag{7.57}$$

Again this equation can be made adaptive using an exponentially decaying forgetting factor λ:

$$\boldsymbol{r}_{yx}(m) = \lambda\boldsymbol{r}_{yx}(m-1) + \boldsymbol{y}(m)x(m) \tag{7.58}$$

For a recursive solution of the least square error Equation (7.58), we need to obtain a recursive time-update formula for the inverse matrix in the form

$$\boldsymbol{R}_{yy}^{-1}(m)=\boldsymbol{R}_{yy}^{-1}(m-1)+Update(m) \tag{7.59}$$

A recursive relation for the matrix inversion is obtained using the following lemma.

The Matrix Inversion Lemma Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two positive-definite $P\times P$ matrices related by

$$\boldsymbol{A}=\boldsymbol{B}^{-1}+\boldsymbol{C}\boldsymbol{D}^{-1}\boldsymbol{C}^{\mathrm{T}} \tag{7.60}$$

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

$$\boldsymbol{A}^{-1}=\boldsymbol{B}-\boldsymbol{B}\boldsymbol{C}\left(\boldsymbol{D}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B} \tag{7.61}$$

This lemma can be proved by multiplying Equation (7.60) and Equation (7.61). 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

$$\boldsymbol{R}_{yy}(m)=\boldsymbol{A} \tag{7.62}$$

$$\lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1)=\boldsymbol{B} \tag{7.63}$$

$$\boldsymbol{y}(m)=\boldsymbol{C} \tag{7.64}$$

$$\boldsymbol{D}=\text{identity matrix} \tag{7.65}$$

Substituting Equations (7.62) and (7.63) in Equation (7.61), we obtain

$$\boldsymbol{R}_{yy}^{-1}(m)=\lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1)-\frac{\lambda^{-2}\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)}{1+\lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)} \tag{7.66}$$

Now define the variables $\boldsymbol{\Phi}(m)$ and $\boldsymbol{k}(m)$ as

$$\boldsymbol{\Phi}_{yy}(m)=\boldsymbol{R}_{yy}^{-1}(m) \tag{7.67}$$

and

$$\boldsymbol{k}(m) = \frac{\lambda^{-1}\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)}{1+\lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{R}_{yy}^{-1}(m-1)\boldsymbol{y}(m)} \tag{7.68}$$

or

$$\boldsymbol{k}(m) = \frac{\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)}{1+\lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)} \tag{7.69}$$

Using Equations (7.67) and (7.69), the recursive equation (7.66) for computing the inverse matrix can be written as

$$\boldsymbol{\Phi}_{yy}(m) = \lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1) - \lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1) \tag{7.70}$$

From Equations (7.69) and (7.70), we have

$$\begin{aligned}\boldsymbol{k}(m) &= \left[\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1) - \lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\right]\boldsymbol{y}(m) \\ &= \boldsymbol{\Phi}_{yy}(m)\boldsymbol{y}(m)\end{aligned} \tag{7.71}$$

Now Equations (7.70) and (7.71) are used in the following to derive the RLS adaptation algorithm.

Recursive Time-update of Filter Coefficients The least square error filter coefficients are

$$\begin{aligned}\boldsymbol{w}(m) &= \boldsymbol{R}_{yy}^{-1}(m)\,\boldsymbol{r}_{yx}(m) \\ &= \boldsymbol{\Phi}_{yy}(m)\boldsymbol{r}_{yx}(m)\end{aligned} \tag{7.72}$$

Substituting the recursive form of the correlation vector in Equation (7.72) yields

$$\begin{aligned}\boldsymbol{w}(m) &= \boldsymbol{\Phi}_{yy}(m)\left[\lambda\boldsymbol{r}_{yx}(m-1) + \boldsymbol{y}(m)x(m)\right] \\ &= \lambda\boldsymbol{\Phi}_{yy}(m)\boldsymbol{r}_{yx}(m-1) + \boldsymbol{\Phi}_{yy}(m)\boldsymbol{y}(m)x(m)\end{aligned} \tag{7.73}$$

Now substitution of the recursive form of the matrix $\boldsymbol{\Phi}_{yy}(m)$ from Equation (7.70) and $\boldsymbol{k}(m)=\boldsymbol{\Phi}(m)\boldsymbol{y}(m)$ from Equation (7.71) in the right-hand side of Equation (7.73) yields

$$\boldsymbol{w}(m)=\left[\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)-\lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\right]\lambda\boldsymbol{r}_{yx}(m-1)+\boldsymbol{k}(m)x(m) \tag{7.74}$$

or

$$\boldsymbol{w}(m)=\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{r}_{yx}(m-1)-\boldsymbol{k}(m)\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{r}_{yx}(m-1)+\boldsymbol{k}(m)x(m) \tag{7.75}$$

Substitution of $\boldsymbol{w}(m-1)=\boldsymbol{\Phi}(m-1)\boldsymbol{r}_{yx}(m-1)$ in Equation (7.75) yields

$$\boldsymbol{w}(m)=\boldsymbol{w}(m-1)-\boldsymbol{k}(m)\left[x(m)-\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{w}(m-1)\right] \tag{7.76}$$

This equation can be rewritten in the following form

$$\boldsymbol{w}(m)=\boldsymbol{w}(m-1)-\boldsymbol{k}(m)e(m) \tag{7.77}$$

Equation (7.77) is a recursive time-update implementation of the least square error Wiener filter.

RLS Adaptation Algorithm

Input signals: $y(m)$ and $x(m)$

Initial values: $\boldsymbol{\Phi}_{yy}(m)=\delta\mathbf{I}$

$\boldsymbol{w}(0)=\boldsymbol{w}_{\mathrm{I}}$

For m = 1,2, ...

Filter gain vector:

$$\boldsymbol{k}(m)=\frac{\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)}{1+\lambda^{-1}\boldsymbol{y}^{\mathrm{T}}(m)\boldsymbol{\Phi}_{yy}(m-1)\boldsymbol{y}(m)} \tag{7.78}$$

Error signal equation:

$$e(m)=x(m)-\boldsymbol{w}^{\mathrm{T}}(m-1)\boldsymbol{y}(m) \tag{7.79}$$

Filter coefficients:

$$\boldsymbol{w}(m)=\boldsymbol{w}(m-1)-\boldsymbol{k}(m)e(m) \tag{7.80}$$

Inverse correlation matrix update:

$$\boldsymbol{\Phi}_{yy}(m)=\lambda^{-1}\boldsymbol{\Phi}_{yy}(m-1)-\lambda^{-1}\boldsymbol{k}(m)\boldsymbol{y}^{T}(m)\boldsymbol{\Phi}_{yy}(m-1) \tag{7.81}$$

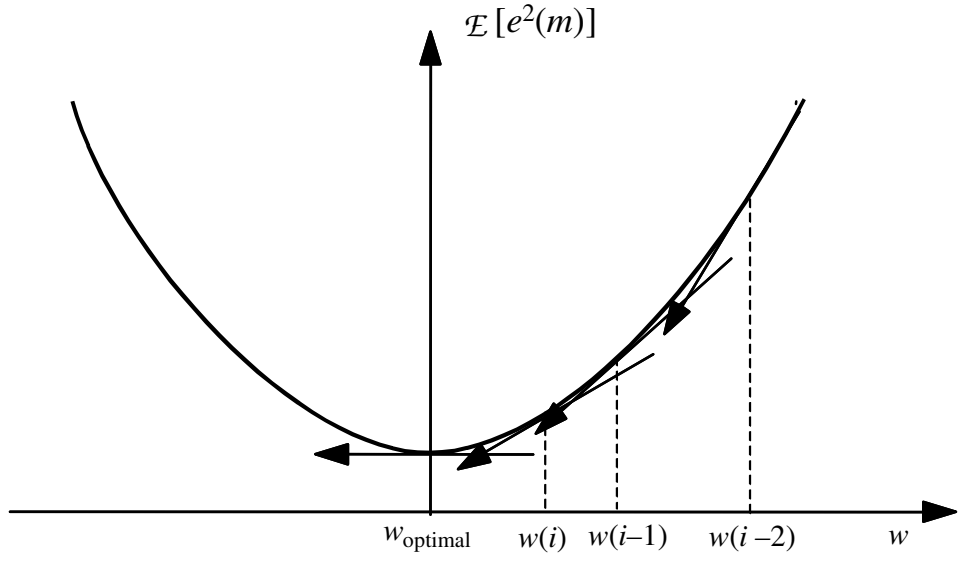


Figure 7.3 Illustration of gradient search of the mean square error surface for the minimum error point.

7.4 The Steepest-Descent Method

The mean square error surface with respect to the coefficients of an FIR filter, is a quadratic bowl-shaped curve, with a single global minimum that corresponds to the LSE filter coefficients. Figure 7.3 illustrates the mean square error curve for a single coefficient filter. This figure also illustrates the steepest-descent search for the minimum mean square error coefficient. The search is based on taking a number of successive downward steps in the direction of negative gradient of the error surface. Starting with a set of initial values, the filter coefficients are successively updated in the downward direction, until the minimum point, at which the gradient is zero, is reached. The steepest-descent adaptation method can be expressed as

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu \left[-\frac{\partial\, \mathcal{E}[e^2(m)]}{\partial\, \boldsymbol{w}(m)} \right] \tag{7.82}$$

where μ is the adaptation step size. From Equation (5.7), the gradient of the mean square error function is given by

$$\frac{\partial\, \mathcal{E}[e^2(m)]}{\partial\, \boldsymbol{w}(m)} = -2\boldsymbol{r}_{yx} + 2\boldsymbol{R}_{yy}\boldsymbol{w}(m) \tag{7.83}$$

Substituting Equation (7.83) in Equation (7.82) yields

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\left[\boldsymbol{r}_{yx} - \boldsymbol{R}_{yy}\boldsymbol{w}(m)\right] \tag{7.84}$$

where the factor of 2 in Equation (7.83) 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

$$\tilde{\boldsymbol{w}}(m) = \boldsymbol{w}(m) - \boldsymbol{w}_o \tag{7.85}$$

For a stationary process, the optimal LSE filter $\boldsymbol{w}_o$ is obtained from the Wiener filter, Equation (5.10), as

$$\boldsymbol{w}_o = \boldsymbol{R}_{yy}^{-1}\boldsymbol{r}_{yx} \tag{7.86}$$

Subtracting $\boldsymbol{w}_o$ from both sides of Equation (7.84), and then substituting $\boldsymbol{R}_{yy}\boldsymbol{w}_o$ for $\boldsymbol{r}_{yx}$, and using Equation (7.85) yields

$$\tilde{\boldsymbol{w}}(m+1) = \left[\mathbf{I} - \mu\boldsymbol{R}_{yy}\right]\tilde{\boldsymbol{w}}(m) \tag{7.87}$$

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 (7.87), 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

$$\boldsymbol{R}_{yy} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}} \tag{7.88}$$

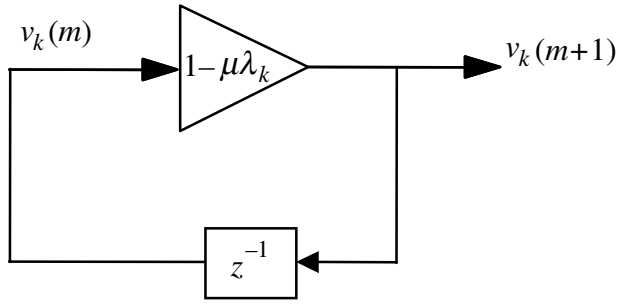


Figure 7.4 A feedback model of the variation of coefficient error with time.

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 (7.88) in Equation (7.87) yields

$$\tilde{\boldsymbol{w}}(m+1)=\left[\mathbf{I}-\mu\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}}\right]\tilde{\boldsymbol{w}}(m) \tag{7.89}$$

Multiplying both sides of Equation (7.89) by $\boldsymbol{Q}^{\mathrm{T}}$ and using the relation $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{Q}\boldsymbol{Q}^{\mathrm{T}}=\mathbf{I}$ yields

$$\boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m+1)=[\mathbf{I}-\mu\boldsymbol{\Lambda}]\boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m) \tag{7.90}$$

Let

$$\boldsymbol{v}(m)=\boldsymbol{Q}^{\mathrm{T}}\tilde{\boldsymbol{w}}(m) \tag{7.91}$$

Then

$$\boldsymbol{v}(m+1)=[\mathbf{I}-\mu\boldsymbol{\Lambda}]\boldsymbol{v}(m) \tag{7.92}$$

As $\boldsymbol{\Lambda}$ and $\mathbf{I}$ are both diagonal matrices, Equation (7.92) can be expressed in terms of the equations for the individual elements of the error vector $\boldsymbol{v}(m)$ as

$$v_k(m+1)=[1-\mu\lambda_k]v_k(m) \tag{7.93}$$

where λ_k is the k^{th} eigenvalue of the autocorrelation matrix of the filter input $y(m)$. Figure 7.4 is a feedback network model of the time variations of the error vector. From Equation (7.93), the condition for the stability of the adaptation process and the decay of the coefficient error vector is

$$-1<1-\mu\lambda_k<1 \tag{7.94}$$

Let λ_{max} denote the maximum eigenvalue of the autocorrelation matrix of $y(m)$ then, from Equation (7.94) the limits on μ for stable adaptation are given by

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{7.95}$$

Convergence Rate The convergence rate of the filter coefficients depends on the choice of the adaptation step size μ, where $0<\mu<1/\lambda_{max}$. When the eigenvalues of the correlation matrix are unevenly spread, the filter coefficients converge at different speeds: the smaller the k^{th} eigenvalue the slower the speed of convergence of the k^{th} coefficients. The filter coefficients with maximum and minimum eigenvalues, λ_{max} and λ_{min} converge according to the following equations:

$$v_{max}(m+1) = (1-\mu\lambda_{max})v_{max}(m) \tag{7.96}$$

$$v_{min}(m+1) = (1-\mu\lambda_{min})v_{min}(m) \tag{7.97}$$

The ratio of the maximum to the minimum eigenvalue of a correlation matrix is called the eigenvalue spread of the correlation matrix:

$$\text{eigenvalue spread} = \frac{\lambda_{max}}{\lambda_{min}} \tag{7.98}$$

Note that the spread in the speed of convergence of filter coefficients is proportional to the spread in eigenvalue of the autocorrelation matrix of the input signal.

7.5 The LMS Filter

The steepest-descent method employs the gradient of the *averaged* squared error to search for the least square error filter coefficients. 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

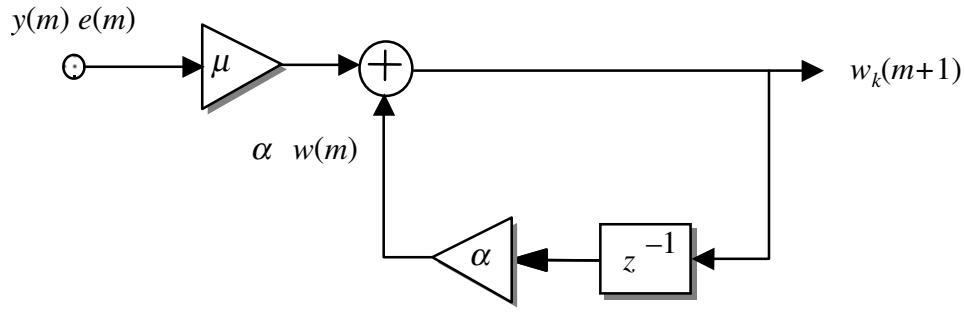


Figure 7.5 Illustration of LMS adaptation of a filter coefficient.

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\left(-\frac{\partial e^2(m)}{\partial \boldsymbol{w}(m)}\right) \tag{7.99}$$

where the error signal $e(m)$ is given by

$$e(m)=x(m) - \boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{x}(m) \tag{7.100}$$

The instantaneous gradient of the squared error can be re-expressed as

$$\begin{aligned}\frac{\partial e^2(m)}{\partial \boldsymbol{w}(m)} &= \frac{\partial}{\partial \boldsymbol{w}(m)}[x(m)-\boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)]^2 \\ &= -2\boldsymbol{y}(m)[x(m)-\boldsymbol{w}^{\mathrm{T}}(m)\boldsymbol{y}(m)]^2 \\ &= -2\boldsymbol{y}(m)e(m)\end{aligned} \tag{7.101}$$

Substituting Equation (7.101) into the recursion update equation of the filter parameters, Equation (7.99) yields the LMS adaptation equation:

$$\boldsymbol{w}(m+1) = \boldsymbol{w}(m) + \mu\,[\boldsymbol{y}(m)e(m)] \tag{7.102}$$

It can 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 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.

Leaky LMS Algorithm The stability and the adaptability of the recursive LMS adaptation Equation (7.86) can improved by introducing a so-called leakage factor α as

$$\boldsymbol{w}(m+1) = \alpha \boldsymbol{w}(m) + \mu \left[\boldsymbol{y}(m) e(m) \right] \tag{7.103}$$

Note that 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. When the parameter $\alpha<1$, the effect is to introduce more stability and accelerate the filter adaptation to the changes in input signal characteristics.

Steady-State Error: The optimal 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 $\mu e(m)\boldsymbol{y}(m)$ 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.

7.6 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 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 (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**, pp. 793–807.

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**, pp. 43–57.

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**, pp. 304–337.

CLASSEN T.A. and MECKLANBRAUKER W.F., (1985) Adaptive Techniques for Signal Processing in Communications. IEEE Communications, **23**, pp. 8–19.

COWAN C.F. and GRANT P.M. (1985) Adaptive Filters. Prentice-Hall, Englewood Cliffs, NJ.

EWEDA E. and MACCHI O. (1985) Tracking Error Bounds of Adaptive Non-sationary Filtering. Automatica, **21**, pp. 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**, pp. 422–38.

GABRIEL W.F. (1976) Adaptive Arrays: An Introduction. Proc. IEEE, **64**, pp. 239–272.

HAYKIN S.(1991) 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.

KAILATH T. (1970) The Innovations Approach to Detection and Estimation Theory, Proc. IEEE, **58**, pp. 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**, pp. 34–45.

KALMAN R.E. and BUCY R.S. (1961) New Results in Linear Filtering and Prediction Theory. Trans. ASME J. Basic Eng., **83**, pp. 95–108.

WIDROW B. (1990) 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Back Propagation. Proc. IEEE, Special Issue on Neural Networks I, **78**.

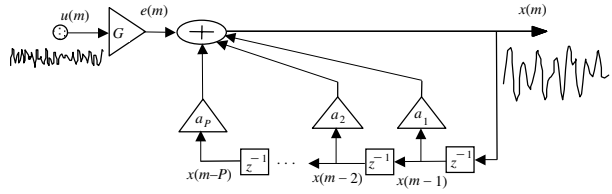
WIDROW B. and STERNS S.D. (1985) Adaptive Signal Processing. Prentice Hall, Englewood Cliffs, NJ.

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.

Advanced Digital Signal Processing and Noise Reduction, Second Edition.
Saeed V. Vaseghi
Copyright © 2000 John Wiley & Sons Ltd
ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

8



LINEAR PREDICTION MODELS

Linear prediction modelling is used in a diverse area of applications, such as data forecasting, speech coding, video coding, speech recognition, model-based spectral analysis, model-based interpolation, signal restoration, and impulse/step event 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 modelling 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 the modelling of signals with a quasi-periodic structure, such as voiced speech, an extended linear predictor that simultaneously utilizes the short and long-term correlation structures is introduced. We study sub-band linear predictors that are particularly useful for sub-band processing of noisy signals. Finally, the application of linear prediction in enhancement of noisy speech is considered. Further applications of linear prediction models in this book are in Chapter 11 on the interpolation of a sequence of lost samples, and in Chapters 12 and 13 on the detection and removal of impulsive noise and transient noise pulses.

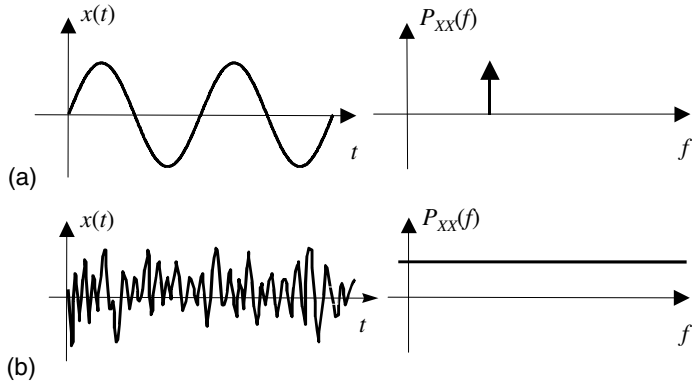


Figure 8.1 The concentration or spread of power in frequency indicates the predictable or random character of a signal: (a) a predictable signal; (b) a random signal.

8.1 Linear Prediction Coding

The success with which a signal can be predicted from its past samples depends on the autocorrelation function, or equivalently the bandwidth and the power spectrum, of the signal. As illustrated in Figure 8.1, in the time domain, a predictable signal has a smooth and correlated fluctuation, and in the frequency domain, the energy of a predictable signal is concentrated in narrow band/s of frequencies. In contrast, the energy of an unpredictable signal, such as a white noise, is spread over a wide band of frequencies.

For a signal to have a capacity to convey information it must have a degree of randomness. Most signals, such as speech, music and video signals, are partially predictable and partially random. These signals can be modelled as the output of a filter excited by an uncorrelated input. The random input models the unpredictable part of the signal, whereas the filter models the predictable structure of the signal. The aim of linear prediction is to model the mechanism that introduces the correlation in a signal.

Linear prediction models are extensively used in speech processing, 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 the vibrations of the glottal cords and the resonance of the vocal tract. Figure 8.2 illustrates a source-filter model of speech. The source models the lung, and emits a random input excitation signal which is filtered by a pitch filter.

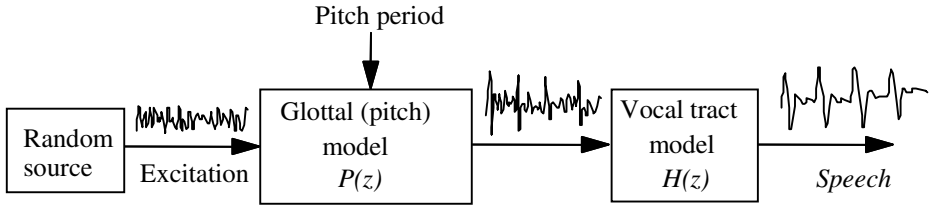


Figure 8.2 A source–filter model of speech production.

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 8.2. The pitch filter model is also termed the "long-term predictor" since it models the correlation of each sample with the samples a pitch period away. The main source of correlation and power in speech is the vocal tract. The vocal tract is modelled by a linear predictor model, which is also termed the "short-term predictor", because it models the correlation of each sample with the few preceding samples. In this section, we study the short-term linear prediction model. In Section 8.3, the predictor model is extended to include long-term pitch period correlations.

A linear predictor model forecasts the amplitude of a signal at time m, $x(m)$, using a linearly weighted combination of P past samples [$x(m-1)$, $x(m-2)$, ..., $x(m-P)$] 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 of Equation (8.1) is illustrated in Figure 8.3.

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

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=1}^{P} a_k x(m-k) \end{aligned} \tag{8.2}$$

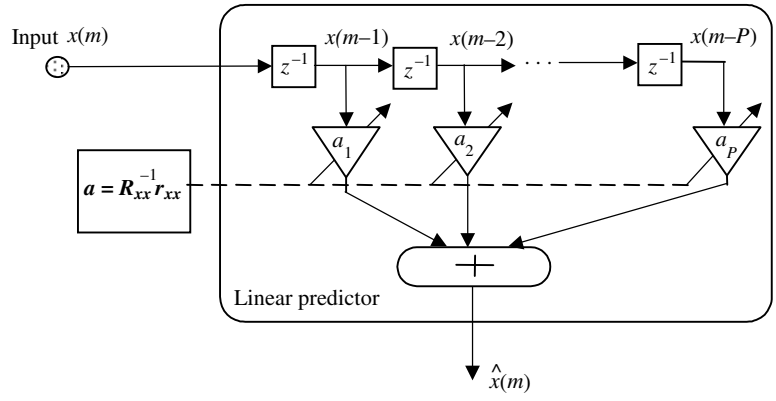


Figure 8.3 Block-diagram illustration of a linear predictor.

For information-bearing signals, the prediction error $e(m)$ may be regarded as the information, or the innovation, content of the sample $x(m)$. From Equation (8.2) a signal generated, or modelled, by a linear predictor can be described by the following feedback equation

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + e(m) \tag{8.3}$$

Figure 8.4 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 of $e(m)$:

$$G=\left(\mathcal{E}\left[e^2(m)\right]\right)^{1/2} \tag{8.4}$$

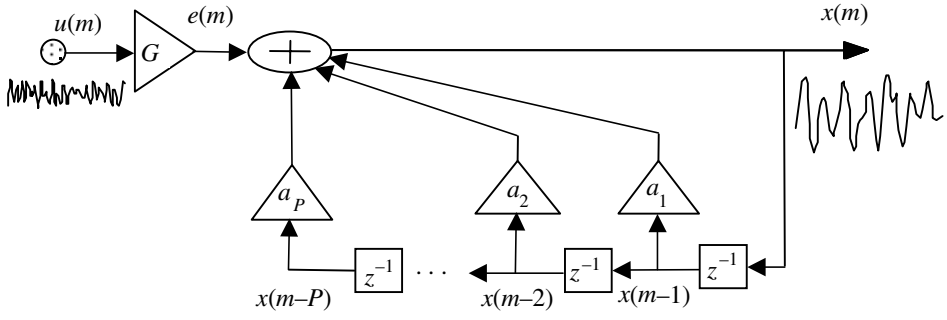


Figure 8.4 Illustration of a signal generated by a linear predictive model.

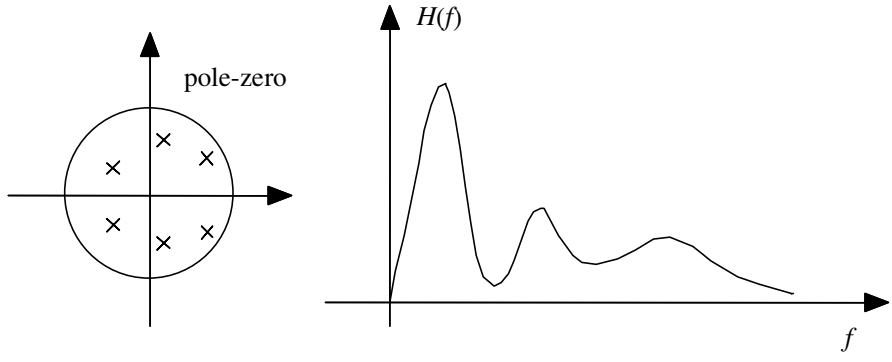


Figure 8.5 The pole–zero position and frequency response of a linear predictor.

where $\mathcal{E}[\cdot]$ is an averaging, or expectation, operator. Taking the z-transform of Equation (8.3) shows that the linear prediction model is an all-pole digital filter with z-transfer function

$$H(z)=\frac{X(z)}{U(z)}=\frac{G}{1-\sum_{k=1}^{P} a_k z^{-k}} \tag{8.5}$$

In general, a linear predictor of order P has $P/2$ complex pole pairs, and can model up to $P/2$ resonance of the signal spectrum as illustrated in Figure 8.5. Spectral analysis using linear prediction models is discussed in Chapter 9.

8.1.1 Least Mean Square Error Predictor

The "best" predictor coefficients are normally obtained by minimising a mean square error criterion defined as

$$\begin{aligned}\mathcal{E}[e^2(m)]&=\mathcal{E}\left[\left(x(m)-\sum_{k=1}^{P} a_k x(m-k)\right)^2\right]\\&=\mathcal{E}[x^2(m)]-2\sum_{k=1}^{P} a_k\,\mathcal{E}[x(m)x(m-k)]+\sum_{k=1}^{P} a_k \sum_{j=1}^{P} a_j\,\mathcal{E}[x(m-k)x(m-j)]\\&=r_{xx}(0)-2\boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{T}}\boldsymbol{a}+\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{a}\end{aligned} \tag{8.6}$$

where $\boldsymbol{R}_{xx} = \mathcal{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]$ is the autocorrelation matrix of the input vector $\boldsymbol{x}^{\mathrm{T}}=[x(m-1), x(m-2), \ldots, x(m-P)]$, $\boldsymbol{r}_{xx}=\mathcal{E}[x(m)\boldsymbol{x}]$ is the autocorrelation vector and $\boldsymbol{a}^{\mathrm{T}}=[a_1, a_2, \ldots, a_P]$ is the predictor coefficient vector. From Equation (8.6), the gradient of the mean square prediction error with respect to the predictor coefficient vector $\boldsymbol{a}$ is given by

$$\frac{\partial}{\partial \boldsymbol{a}}\mathcal{E}[e^2(m)] = -2\boldsymbol{r}_{xx}^{\mathrm{T}} + 2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{R}_{xx} \tag{8.7}$$

where the gradient vector is defined as

$$\frac{\partial}{\partial \boldsymbol{a}} = \left(\frac{\partial}{\partial a_1}, \frac{\partial}{\partial a_2}, \ldots, \frac{\partial}{\partial a_{\mathrm{P}}}\right)^{\mathrm{T}} \tag{8.8}$$

The least mean square error solution, obtained by setting Equation (8.7) to zero, is given by

$$\boldsymbol{R}_{xx}\,\boldsymbol{a} = \boldsymbol{r}_{xx} \tag{8.9}$$

From Equation (8.9) the predictor coefficient vector is given by

$$\boldsymbol{a} = \boldsymbol{R}_{xx}^{-1}\boldsymbol{r}_{xx} \tag{8.10}$$

Equation (8.10) may also be written in an expanded form as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \cdots & r_{xx}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{xx}(1) \\ r_{xx}(2) \\ r_{xx}(3) \\ \vdots \\ r_{xx}(P) \end{pmatrix} \tag{8.11}$$

An alternative formulation of the least square error problem is as follows. For a signal block of N samples $[x(0), \ldots, x(N-1)]$, we can write a set of N linear prediction error equations as

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(-1) & x(-2) & x(-3) & \cdots & x(-P) \\ x(0) & x(-1) & x(-2) & \cdots & x(1-P) \\ x(1) & x(0) & x(-1) & \cdots & x(2-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(N-2) & x(N-3) & x(N-4) & \cdots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{8.12}$$

where $\boldsymbol{x}^{\mathrm{T}} = [x(-1), ..., x(-P)]$ is the initial vector. In a compact vector/matrix notation Equation (8.12) can be written as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{X}\boldsymbol{a} \tag{8.13}$$

Using Equation (8.13), the sum of squared prediction errors over a block of N samples can be expressed as

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} - 2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{a} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{a} \tag{8.14}$$

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:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{a}} = -2\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} = 0 \tag{8.15}$$

From Equation (8.15), the least square error predictor is given by

$$\boldsymbol{a} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}\right) \tag{8.16}$$

A comparison of Equations (8.11) and (8.16) shows that in Equation (8.16) the autocorrelation matrix and vector of Equation (8.11) are replaced by the time-averaged estimates as

$$\hat{r}_{xx}(m) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)x(k-m) \tag{8.17}$$

Equations (8.11) and (8.16) 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 [$r_{xx}(0)$, $r_{xx}(1)$, . . . , $r_{xx}(P)$] in the correlation matrix and the cross-correlation vector. An efficient method for solution of Equation (8.10) is the Levinson–Durbin algorithm, introduced in Section 8.2.2.

8.1.2 The Inverse Filter: Spectral Whitening

The all-pole linear predictor model, in Figure 8.4, 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 8.6 is given by

$$X(f)=\frac{G\,U(f)}{A(f)}=\frac{E(f)}{1-\sum_{k=1}^{P} a_k\, e^{-\mathrm{j}2\pi fk}} \tag{8.18}$$

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,

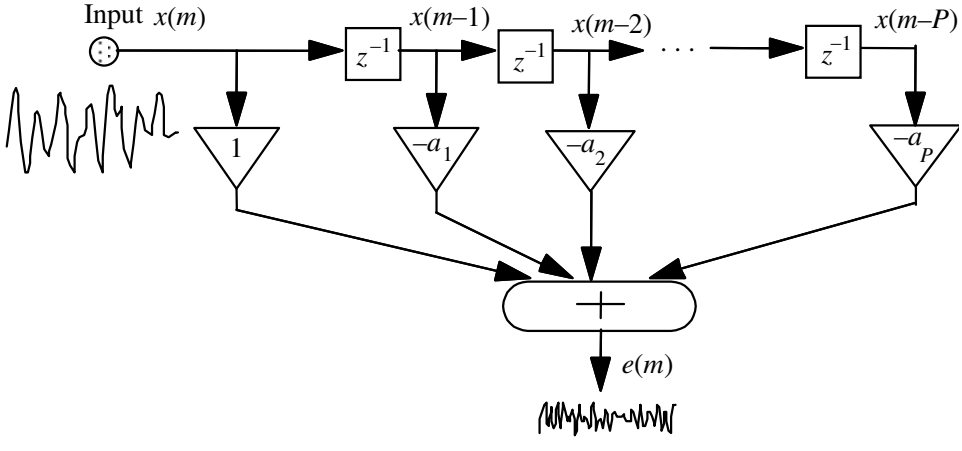


Figure 8.6 Illustration of the inverse (or whitening) filter.

as the name implies, transforms a correlated signal $x(m)$ back to an uncorrelated flat-spectrum signal $e(m)$. The inverse filter, also known as the prediction error filter, is an all-zero finite impulse response filter defined as

$$\begin{aligned} e(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=1}^{P} a_k x(m-k) \\ &= (\boldsymbol{a}^{\text{inv}})^{\text{T}} \boldsymbol{x} \end{aligned} \tag{8.19}$$

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

$$A(z) = 1 - \sum_{k=1}^{P} a_k \, z^{-k} \tag{8.20}$$

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 8.7. Consequently, the zeros of the inverse filter introduce anti-resonances that cancel out the resonances of the poles of the predictor. The inverse filter has the effect of flattening the spectrum of the input signal, and is also known as a spectral whitening, or decorrelation, filter.

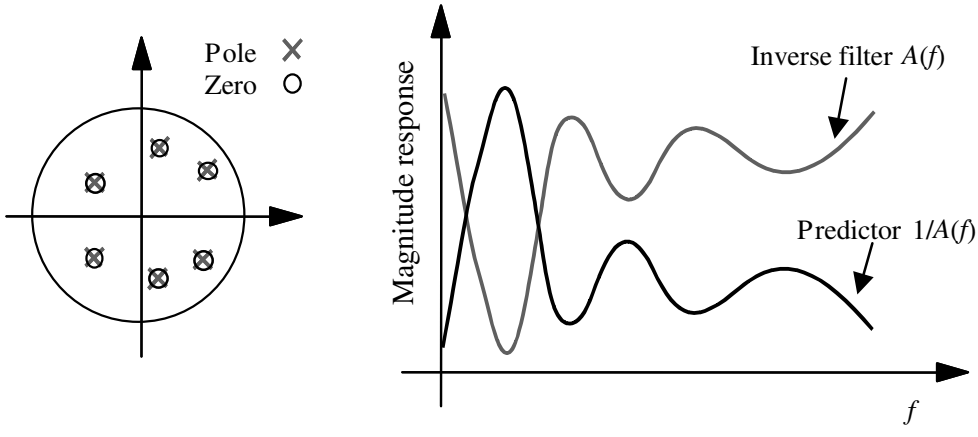


Figure 8.7 Illustration of the pole-zero diagram, and the frequency responses of an all-pole predictor and its all-zero inverse filter.

8.1.3 The Prediction Error Signal

The prediction error signal is in general composed of three components:

(a) the input signal, also called the excitation signal;
(b) the errors due to the modelling inaccuracies;
(c) 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 nonzero because information bearing signals are random, often only approximately modelled by a linear system, and usually observed in noise. The least mean square prediction error, obtained from substitution of Equation (8.9) in Equation (8.6), is

$$E^{(P)} = \mathcal{E}[e^2(m)] = r_{xx}(0) - \sum_{k=1}^{P} a_k r_{xx}(k) \tag{8.21}$$

where $E^{(P)}$ denotes the prediction error for a predictor of order P. The prediction error decreases, initially rapidly and then slowly, with 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

$$\mathcal{E}[e(m)e(m-k)] = \begin{cases} \sigma_e^2 = G^2 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases} \tag{8.22}$$

where σ_e^2 is the variance of $e(m)$.

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 $x(m-1)$, $x(m-2)$, . . .,$x(m-P)$.

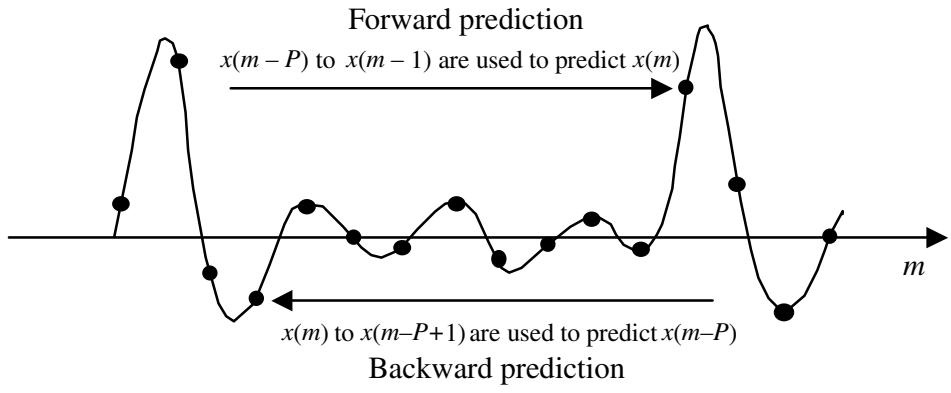


Figure 8.8 Illustration of forward and backward predictors.

Similarly, as shown in Figure 8.8, we can define a backward predictor, that predicts a sample $x(m-P)$ from P *future* samples $x(m-P+1), \ldots, x(m)$ as

$$\hat{x}(m-P)=\sum_{k=1}^{P} c_k\, x(m-k+1) \tag{8.23}$$

The backward prediction error is defined as the difference between the actual sample and its predicted value:

$$\begin{aligned} b(m) &= x(m-P)-\hat{x}(m-P) \\ &= x(m-P)-\sum_{k=1}^{P} c_k\, x(m-k+1) \end{aligned} \tag{8.24}$$

From Equation (8.24), a signal generated by a backward predictor is given by

$$x(m-P)=\sum_{k=1}^{P} c_k\, x(m-k+1)+b(m) \tag{8.25}$$

The coefficients of the least square error backward predictor, obtained in a similar method to that of the forward predictor in Section 8.1.1, are given by

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \dots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \dots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_P \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \tag{8.26}$$

Note that the main difference between Equations (8.26) and (8.11) is that the correlation vector on the right-hand side of the backward predictor, Equation (8.26) is upside-down compared with the forward predictor, Equation (8.11). Since the correlation matrix is Toeplitz and symmetric, Equation (8.11) for the forward predictor may be rearranged and rewritten in the following form:

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \dots & r_{xx}(P-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(P-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(P-1) & r_{xx}(P-2) & r_{xx}(P-3) & \dots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_P \\ a_{P-1} \\ a_{P-2} \\ \vdots \\ a_1 \end{pmatrix} = \begin{pmatrix} r_{xx}(P) \\ r_{xx}(P-1) \\ r_{xx}(P-2) \\ \vdots \\ r_{xx}(1) \end{pmatrix} \tag{8.27}$$

A comparison of Equations (8.27) and (8.26) shows that the coefficients of the backward predictor are the time-reversed versions of those of the forward predictor

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_P \end{pmatrix} = \begin{pmatrix} a_P \\ a_{P-1} \\ a_{P-2} \\ \vdots \\ a_1 \end{pmatrix} = \boldsymbol{a}^{\mathrm{B}} \tag{8.28}$$

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 $[1, -a_1, ..., -a_P]=[1, -\boldsymbol{a}^{\mathrm{T}}]$. Equations (8.11) and (8.21) may be combined to yield a matrix equation for the inverse forward predictor coefficients:

$$\begin{pmatrix} r(0) & \boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{T}} \\ \boldsymbol{r}_{\boldsymbol{xx}} & \boldsymbol{R}_{\boldsymbol{xx}} \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a} \end{pmatrix} = \begin{pmatrix} E^{(P)} \\ \mathbf{0} \end{pmatrix} \tag{8.29}$$

Equation (8.29) is called the augmented forward predictor equation. Similarly, for the inverse backward predictor, we can define an augmented backward predictor equation as

$$\begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}} & \boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{B}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{BT}} & r(0) \end{pmatrix} \begin{pmatrix} -\boldsymbol{a}^{\mathrm{B}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ E^{(P)} \end{pmatrix} \tag{8.30}$$

where $\boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{T}} = [r_{xx}(1), \cdots, r_{xx}(P)]$ and $\boldsymbol{r}_{\boldsymbol{xx}}^{\mathrm{BT}} = [r_{xx}(P), \ldots, r_{xx}(1)]$. Note that the superscript BT denotes backward and transposed. The augmented forward and backward matrix Equations (8.29) and (8.30) 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 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

$$\begin{pmatrix} 1 \\ -a_1^{(i)} \\ \vdots \\ -a_{i-1}^{(i)} \\ -a_i^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -a_1^{(i-1)} \\ \vdots \\ -a_{i-1}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -a_{i-1}^{(i-1)} \\ \vdots \\ -a_1^{(i-1)} \\ 1 \end{pmatrix} \tag{8.31}$$

or in a more compact vector notation as

$$\begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i)} \end{pmatrix} = \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} \tag{8.32}$$

where k_i is called the reflection coefficient. The proof of Equation (8.32) 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

$$\begin{pmatrix} -\boldsymbol{a}^{(i)\mathrm{B}} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} + k_i \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} \tag{8.33}$$

To prove the order-update Equation (8.32) (or alternatively Equation (8.33)), 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

$$\boldsymbol{R}_{\boldsymbol{xx}}^{(i+1)} = \begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{BT}} & r_{xx}(0) \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{T}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} \end{pmatrix} \tag{8.34}$$

to obtain

$$\begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{BT}} & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{B}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{BT}} & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\boldsymbol{a}^{(i-1)} \\ 0 \end{pmatrix} + k_i \begin{pmatrix} r_{xx}(0) & \boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{T}} \\ \boldsymbol{r}_{\boldsymbol{xx}}^{(i)} & \boldsymbol{R}_{\boldsymbol{xx}}^{(i)} \end{pmatrix} \begin{pmatrix} 0 \\ -\boldsymbol{a}^{(i-1)\mathrm{B}} \\ 1 \end{pmatrix} \tag{8.35}$$

where in Equation (8.34) and Equation (8.35) $\boldsymbol{r}_{\boldsymbol{xx}}^{(i)\mathrm{T}} = [r_{xx}(1), \cdots, r_{xx}(i)]$, 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 (8.35) and the use of Equations (8.29) and (8.30) yields

$$\begin{pmatrix} E^{(i)} \\ \mathbf{0}^{(i)} \end{pmatrix} = \begin{pmatrix} E^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ \Delta^{(i-1)} \end{pmatrix} + k_i \begin{pmatrix} \Delta^{(i-1)} \\ \mathbf{0}^{(i-1)} \\ E^{(i-1)} \end{pmatrix} \tag{8.36}$$

where

$$\begin{aligned} \Delta^{(i-1)} &= \begin{bmatrix} 1 & -\boldsymbol{a}^{(i-1)} \end{bmatrix}^{\mathrm{T}} \boldsymbol{r}_{xx}^{(i)\mathrm{B}} \\ &= r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \end{aligned} \tag{8.37}$$

If Equation (8.36) is true, it follows that Equation (8.32) must also be true. The conditions for Equation (8.36) to be true are

$$E^{(i)} = E^{(i-1)} + k_i \Delta^{(i-1)} \tag{8.38}$$

and

$$0 = \Delta^{(i-1)} + k_i E^{(i-1)} \tag{8.39}$$

From (8.39),

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \tag{8.40}$$

Substitution of $\Delta^{(i-1)}$ from Equation (8.40) into Equation (8.38) yields

$$\begin{aligned} E^{(i)} &= E^{(i-1)}(1-k_i^2) \\ &= E^{(0)} \prod_{j=1}^{i} (1-k_j^2) \end{aligned} \tag{8.41}$$

Note that it can be shown that $\Delta^{(i)}$ is the cross-correlation of the forward and backward prediction errors:

$$\Delta^{(i-1)} = \mathcal{E}[b^{(i-1)}(m-1)e^{(i-1)}(m)] \tag{8.42}$$

The parameter $\Delta^{(i-1)}$ is known as the partial correlation.

Durbin's algorithm

Equations (8.43)–(8.48) are solved recursively for i=1, . . ., P. 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:

$$E^{(0)} = r_{xx}(0) \tag{8.43}$$

For i =1, ..., P

$$\Delta^{(i-1)} = r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k) \tag{8.44}$$

$$k_i = -\frac{\Delta^{(i-1)}}{E^{(i-1)}} \tag{8.45}$$

$$a_i^{(i)} = k_i \tag{8.46}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i\, a_{i-j}^{(i-1)} \qquad 1 \le j \le i-1 \tag{8.47}$$

$$E^{(i)} = (1-k_i^2)E^{(i-1)} \tag{8.48}$$

8.2.3 Lattice Predictors

The lattice structure, shown in Figure 8.9, 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 (8.32) by the input vector [$x(m)$, $x(m-1)$, . . . , $x(m-i)$]:

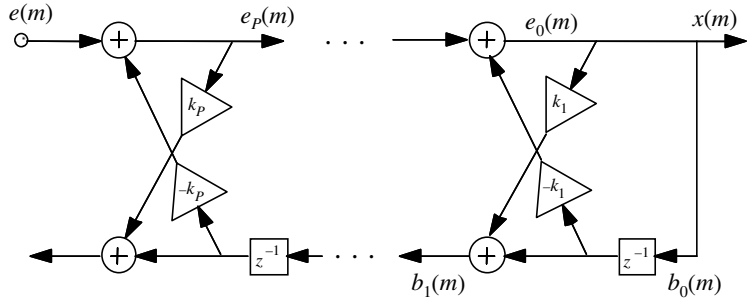
$$e^{(i)}(m) = e^{(i-1)}(m) - k_i\, b^{(i-1)}(m-1) \tag{8.49}$$

Similarly, we can obtain an order-update recursive equation for the backward prediction error by multiplying both sides of Equation (8.33) by the input vector $[x(m–i), x(m–i+1), \ldots, x(m)]$ as

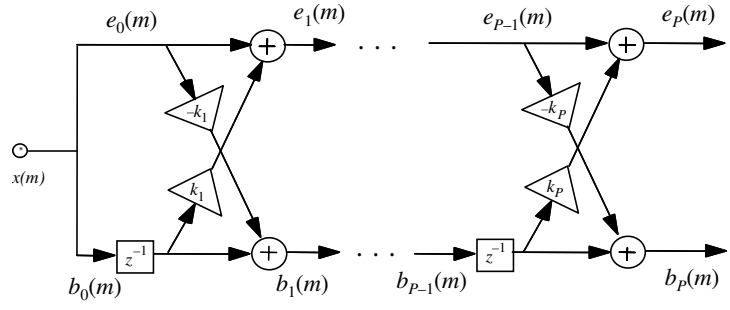
$$b^{(i)}(m)=b^{(i-1)}(m-1)-k_i e^{(i-1)}(m) \tag{8.50}$$

Equations (8.49) and (8.50) are interrelated and may be implemented by a lattice network as shown in Figure 8.8. Minimisation of the squared forward prediction error of Equation (8.49) over N samples yields

$$k_i=\frac{\sum_{m=0}^{N-1} e^{(i-1)}(m)b^{(i-1)}(m-1)}{\sum_{m=0}^{N-1}\left(e^{(i-1)}(m)\right)^2} \tag{8.51}$$



(a)



(b)

Figure 8.9 Configuration of (a) a lattice predictor and (b) the inverse lattice predictor.

Note that a similar relation for k_i can be obtained through minimisation of the squared backward prediction error of Equation (8.50) over N samples. The reflection coefficients are also known as the normalised partial correlation (PARCOR) coefficients.

8.2.4 Alternative Formulations of Least Square Error Prediction

The methods described above for derivation of the predictor coefficients are based on minimisation of either the forward or the backward prediction error. In this section, we consider alternative methods based on the minimisation of the sum of the forward and backward prediction errors.

Burg's Method Burg's method is based on minimisation of the sum of the forward and backward squared prediction errors. The squared error function is defined as

$$E_{fb}^{(i)} = \sum_{m=0}^{N-1} \left\{ \left[e^{(i)}(m) \right]^2 + \left[b^{(i)}(m) \right]^2 \right\} \tag{8.52}$$

Substitution of Equations (8.49) and (8.50) in Equation (8.52) yields

$$E_{fb}^{(i)} = \sum_{m=0}^{N-1} \left\{ \left[e^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \right]^2 + \left[b^{(i-1)}(m-1) - k_i e^{(i-1)}(m) \right]^2 \right\} \tag{8.53}$$

Minimisation of $E_{fb}^{(i)}$ with respect to the reflection coefficients k_i yields

$$k_i = \frac{2\sum_{m=0}^{N-1} e^{(i-1)}(m) b^{(i-1)}(m-1)}{\sum_{m=0}^{N-1} \left\{ \left[e^{(i-1)}(m) \right]^2 + \left[b^{(i-1)}(m-1) \right]^2 \right\}} \tag{8.54}$$

Simultaneous Minimisation of the Backward and Forward Prediction Errors From Equation (8.28) 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 the following equation:

$$\begin{aligned} E_{fb}^{(P)} &= \sum_{m=0}^{N-1}\left\{\left[e^{(P)}(m)\right]^2+\left[b^{(P)}(m)\right]^2\right\} \\ &= \sum_{m=0}^{N-1}\left\{\left[x(m)-\sum_{k=1}^{P}a_k x(m-k)\right]^2+\left[x(m-P)-\sum_{k=1}^{P}a_k x(m-P+k)\right]^2\right\} \\ &= (\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})+\left(\boldsymbol{x}^{\mathrm{B}}-\boldsymbol{X}^{\mathrm{B}}\boldsymbol{a}\right)^{\mathrm{T}}\left(\boldsymbol{x}^{\mathrm{B}}-\boldsymbol{X}^{\mathrm{B}}\boldsymbol{a}\right) \end{aligned} \tag{8.55}$$

where $\boldsymbol{X}$ and $\boldsymbol{x}$ are the signal matrix and vector defined by Equations (8.12) and (8.13), 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 (8.16), the minimisation of the mean squared error function of Equation (8.54) yields

$$\boldsymbol{a}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{X}^{\mathrm{BT}}\boldsymbol{X}^{\mathrm{B}}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{X}^{\mathrm{BT}}\boldsymbol{x}^{\mathrm{B}}\right) \tag{8.56}$$

Note that for an ergodic signal as the signal length N increases Equation (8.56) converges to the so-called normal Equation (8.10).

8.2.5 Predictor Model Order Selection

One procedure for the determination of the correct model order is to increment the model order, and monitor the differential change in the error power, until the change levels off. The incremental change in error power with the increasing model order from i–1 to i is defined as

$$\Delta E^{(i)}=E^{(i-1)}-E^{(i)} \tag{8.57}$$

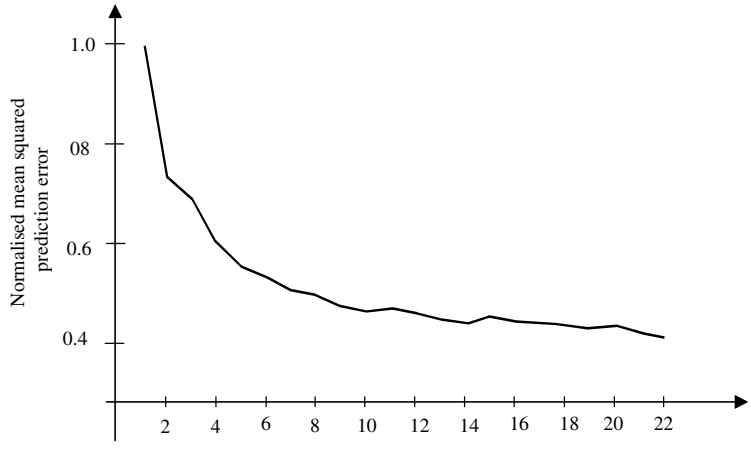


Figure 8.10 Illustration of the decrease in the normalised mean squared prediction error with the increasing predictor length for a speech signal.

Figure 8.10 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 resonances in the spectrum needs $P=2K$ coefficients. Hence a procedure for model selection is to examine the power spectrum of the signal process, and to set the model order to twice the number of significant spectral peaks in the spectrum.

When the model order is less than the correct order, the signal is under-modelled. In this case the prediction error is not well decorrelated 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, there are two types of correlation structures that can be utilised for a more accurate prediction, these are:

(a) the short-term correlation, which is the correlation of each sample with the P immediate past samples: $x(m-1), \ldots, x(m-P)$;
(b) 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 8.11 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

$$\hat{e}(m) = \sum_{k=-Q}^{Q} p_k \, e(m-T-k) \tag{8.58}$$

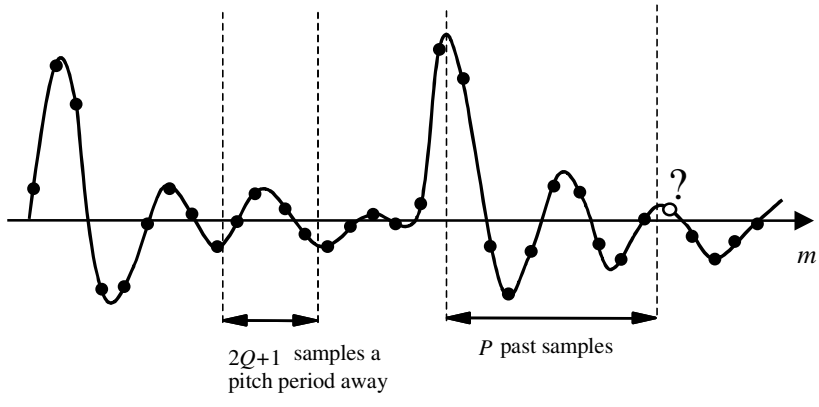


Figure 8.11 Illustration of the short-term relation of a sample with the P immediate past samples and the long-term relation with the samples a pitch period away.

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

$$\begin{aligned}\varepsilon(m) &= e(m) - \hat{e}(m) \\ &= e(m) - \sum_{k=-Q}^{Q} p_k\, e(m-T-k)\end{aligned} \tag{8.59}$$

Minimisation of $\mathcal{E}[e^2(m)]$ results in the following solution for the pitch predictor:

$$\begin{pmatrix} p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{Q-1} \\ p_Q \end{pmatrix} = \begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(2Q) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(2Q-1) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(2Q-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{xx}(2Q) & r_{xx}(2Q-1) & r_{xx}(2Q-2) & \cdots & r_{xx}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{xx}(T-Q) \\ r_{xx}(T-Q+1) \\ \vdots \\ r_{xx}(T+Q-1) \\ r_{xx}(T+Q) \end{pmatrix} \tag{8.60}$$

An alternative to the separate, cascade, modelling of the short- and long-term correlations is to combine the short- and long-term predictors into a single model described as

$$x(m) = \underbrace{\sum_{k=1}^{P} a_k\, x(m-k)}_{\text{short term prediction}} + \underbrace{\sum_{k=-Q}^{Q} p_k\, x(m-k-T)}_{\text{long term prediction}} + \varepsilon(m) \tag{8.61}$$

In Equation (8.61), each sample is expressed as a linear combination of P immediate past samples and $2Q+1$ samples a pitch period away. Minimisation of $\mathcal{E}[e^2(m)]$ results in the following solution for the pitch predictor:

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{+Q} \end{pmatrix} = \begin{pmatrix} r(0) & r(1) & \dots & r(P-1) & r(T+Q-1) & r(T+Q) & \dots & r(T-Q-1) \\ r(1) & r(0) & \dots & r(P-2) & r(T+Q-2) & r(T+Q-1) & \dots & r(T+Q-2) \\ r(2) & r(1) & \dots & r(P-3) & r(T+Q-3) & r(T+Q-2) & \dots & r(T+Q-3) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \dots & r(0) & r(T+Q-P) & r(T+Q-P+1) & \dots & r(T+Q-P) \\ r(T+Q-1) & r(T+Q-2) & \dots & r(T+Q-P) & r(0) & r(1) & \dots & r(2Q) \\ r(T+Q) & r(T+Q-1) & \dots & r(T+Q-P+1) & r(1) & r(0) & \dots & r(2Q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(T-Q-1) & r(T-Q-2) & \cdots & r(T-Q-P) & r(2Q) & r(2Q-1) & \dots & r(0) \end{pmatrix}^{-1} \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T+Q) \\ r(T+Q-1) \\ \vdots \\ r(T-Q) \end{pmatrix} \tag{8.62}$$

In Equation (8.62), for simplicity the subscript xx of $r_{xx}(k)$ has been omitted. In Chapter 10, the predictor model of Equation (8.61) is used for interpolation of a sequence of missing samples.

8.4 MAP Estimation of Predictor Coefficients

The posterior probability density function of a predictor coefficient vector $\boldsymbol{a}$, given a signal $\boldsymbol{x}$ and the initial samples $\boldsymbol{x}_\mathrm{I}$, can be expressed, using Bayes' rule, as

$$f_{\boldsymbol{A}|\boldsymbol{X},\boldsymbol{X}_\mathrm{I}}(\boldsymbol{a} \mid \boldsymbol{x},\boldsymbol{x}_\mathrm{I}) = \frac{f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{x} \mid \boldsymbol{a},\boldsymbol{x}_\mathrm{I}) f_{\boldsymbol{A}|\boldsymbol{X}_\mathrm{I}}(\boldsymbol{a} \mid \boldsymbol{x}_\mathrm{I})}{f_{\boldsymbol{X}|\boldsymbol{X}_I}(\boldsymbol{x} \mid \boldsymbol{x}_\mathrm{I})} \tag{8.63}$$

In Equation (8.63), the pdfs are conditioned on P initial signal samples $\boldsymbol{x}_\mathrm{I}=[x(-P),\ x(-P+1),\ \ldots,\ x(-1)]$. Note that for a given set of samples $[\boldsymbol{x},\ \boldsymbol{x}_\mathrm{I}]$, $f_{\boldsymbol{X}|\boldsymbol{X}_\mathrm{I}}(\boldsymbol{x} \mid \boldsymbol{x}_\mathrm{I})$ is a constant, and it is reasonable to assume that $f_{\boldsymbol{A}|\boldsymbol{X}_I}(\boldsymbol{a} \mid \boldsymbol{x}_\mathrm{I}) = f_{\boldsymbol{A}}(\boldsymbol{a})$.

8.4.1 Probability Density Function of Predictor Output

The pdf $f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{X}_\mathrm{I}}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_\mathrm{I})$ 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 $\boldsymbol{e}$:

$$f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{X}_\mathrm{I}}(\boldsymbol{x} \mid \boldsymbol{a},\boldsymbol{x}_\mathrm{I}) = f_{\boldsymbol{E}}(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}) \tag{8.64}$$

where the input signal vector is given by

$$\boldsymbol{e} = -\boldsymbol{X}\boldsymbol{a} \tag{8.65}$$

and $f_E(\boldsymbol{e})$ is the pdf of $\boldsymbol{e}$. Equation (8.64) can be expanded as

$$\begin{pmatrix} e(0) \\ e(1) \\ e(2) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(-1) & x(-2) & x(-3) & \cdots & x(-P) \\ x(0) & x(-1) & x(-2) & \cdots & x(1-P) \\ x(1) & x(0) & x(-1) & \cdots & x(2-P) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(N-2) & x(N-3) & x(N-4) & \cdots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{8.66}$$

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 (8.64) becomes

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{x} \mid \boldsymbol{a}, \boldsymbol{x}_I) &= f_{\boldsymbol{E}}(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}) \\ &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(\frac{1}{2\sigma_e^2} (\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a})^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}) \right) \end{aligned} \tag{8.67}$$

An alternative form of Equation (8.67) can be obtained by rewriting Equation (8.66) in the following form:

$$\begin{pmatrix} e_0 \\ e_1 \\ e_3 \\ e_4 \\ \vdots \\ e_{N-1} \end{pmatrix} = \begin{pmatrix} -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -a_P & \cdots & -a_2 & -a_1 & 1 \end{pmatrix} \begin{pmatrix} x_{-P} \\ x_{-P+1} \\ x_{-P+2} \\ x_{-P+3} \\ \vdots \\ x_{N-1} \end{pmatrix} \tag{8.68}$$

In a compact notation Equation (8.68) can be written as

$$\boldsymbol{e} = \boldsymbol{A}\boldsymbol{x} \tag{8.69}$$

Using Equation (8.69), and assuming that the excitation signal $e(m)$ is a zero mean, uncorrelated process with variance σ_e^2, the likelihood function of Equation (8.67) can be written as

$$f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{x}_{\mathrm{I}})=\frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}}\exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \tag{8.70}$$

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 $\boldsymbol{\mu}_{\boldsymbol{a}}$ and a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{aa}}$:

$$f_{\boldsymbol{A}}(\boldsymbol{a})=\frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{\boldsymbol{aa}}|^{1/2}}\exp\left[-\frac{1}{2}(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})\right] \tag{8.71}$$

Substituting Equations (8.67) and (8.71) in Equation (8.63), the posterior pdf of the predictor coefficient vector $f_{\boldsymbol{A}|\boldsymbol{X},\boldsymbol{X}_{\mathrm{I}}}(\boldsymbol{a}\mid\boldsymbol{x},\boldsymbol{x}_{\mathrm{I}})$ can be expressed as

$$\begin{aligned}f_{\boldsymbol{A}|\boldsymbol{X},\boldsymbol{X}_I}(\boldsymbol{a}\mid\boldsymbol{x},\boldsymbol{x}_{\mathrm{I}})=&\frac{1}{f_{\boldsymbol{X}|\boldsymbol{X}_{\mathrm{I}}}(\boldsymbol{x}\,|\boldsymbol{x}_{\mathrm{I}})}\frac{1}{(2\pi)^{(N+P)/2}\sigma_e^N|\boldsymbol{\Sigma}_{\boldsymbol{aa}}|^{1/2}}\\&\times\exp\left\{-\frac{1}{2}\left[\frac{1}{\sigma_e^2}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})+(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})\right]\right\}\end{aligned} \tag{8.72}$$

The maximum a posteriori estimate is obtained by maximising the log-likelihood function:

$$\frac{\partial}{\partial\boldsymbol{a}}\left[\ln f_{\boldsymbol{A}|\boldsymbol{X},\boldsymbol{X}_I}(\boldsymbol{a}\mid\boldsymbol{x},\boldsymbol{x}_I)\right]=\frac{\partial}{\partial\boldsymbol{a}}\left[\frac{1}{\sigma_e^2}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{X}\boldsymbol{a})+(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{aa}}^{-1}(\boldsymbol{a}-\boldsymbol{\mu}_{\boldsymbol{a}})\right]=0 \tag{8.73}$$

This yields

$$\hat{\boldsymbol{a}}^{MAP}=\left(\boldsymbol{\Sigma}_{\boldsymbol{aa}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{aa}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}+\sigma_e^2\left(\boldsymbol{\Sigma}_{\boldsymbol{aa}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{\mu}_{\boldsymbol{a}} \tag{8.74}$$

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:

$$\hat{\boldsymbol{a}}^{LS} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x}\right) \tag{8.75}$$

Similarly as the observation length N increases the signal matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ becomes more significant than $\boldsymbol{\Sigma}_{aa}$ and again the MAP solution tends to a least squared error solution.

8.5 Sub-Band Linear Prediction Model

In a P^{th} order linear prediction model, the P predictor coefficients model the signal spectrum over its full spectral bandwidth. The distribution of the LP parameters (or equivalently the poles of the LP model) over the signal bandwidth depends on the signal correlation and spectral structure. Generally, the parameters redistribute themselves over the spectrum to minimize the mean square prediction error criterion. An alternative to a conventional LP model is to divide the input signal into a number of sub-bands and to model the signal within each sub-band with a linear prediction model as shown in Figure 8.12. The advantages of using a sub-band LP model are as follows:

(1) Sub-band linear prediction allows the designer to allocate a specific number of model parameters to a given sub-band. Different numbers of parameters can be allocated to different bands.
(2) The solution of a full-band linear predictor equation, i.e. Equation (8.10) or (8.16), requires the inversion of a relatively large correlation matrix, whereas the solution of the sub-band LP models require the inversion of a number of relatively small 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, …,N. The k^{th} sub-band signal is modelled using a low-order linear prediction model as

$$x_k(m)=\sum_{i=1}^{P_k} a_k(i)x_k(m-i)+g_k e_k(m) \tag{8.76}$$

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

$$P_{EE}(f,k)=\begin{cases}1 & f_{k,start}<f<f_{k,end}\\ 0 & \text{otherwise}\end{cases} \tag{8.77}$$

where $f_{k,start}$, $f_{k,end}$ 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

$$r_{ee}(m)=B_k \operatorname{sinc}[m(B_k-f_{k0})/2] \tag{8.78}$$

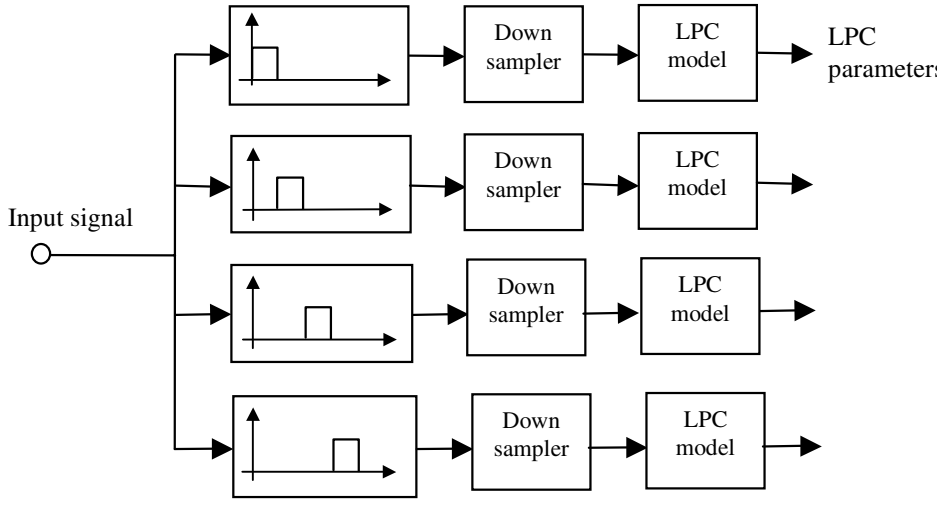


Figure 8.12 Configuration of a sub-band linear prediction model.

where B_k and f_{k0} are the bandwidth and the centre frequency of the k^{th} sub-band respectively. To ensure that each sub-band LP parameters only model the signal within that sub-band, the sub-band signals are down-sampled as shown in Figure 8.12.

8.6 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

$$\begin{aligned} y(m) &= x(m) + n(m) \\ &= \sum_{k=1}^{P} a_k x(m-k) + e(m) + n(m) \end{aligned} \tag{8.79}$$

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

$$\boldsymbol{R}_{yy}\hat{\boldsymbol{a}} = \boldsymbol{r}_{yy} \tag{8.80}$$

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 (8.80) can be written as

$$(\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})(\boldsymbol{a} + \tilde{\boldsymbol{a}}) = (\boldsymbol{r}_{xx} + \boldsymbol{r}_{nn}) \tag{8.81}$$

where $\tilde{\boldsymbol{a}}$ is the error in the predictor coefficients vector due to the noise. A simple method for removing the effects of noise is to subtract an estimate of the autocorrelation of the noise from that of the noisy signal. The drawback

of this approach is that, owing to random variations of noise, correlation subtraction can cause numerical instability in Equation (8.80) and result in spurious solutions. In the following, we formulate the p.d.f. 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 $\boldsymbol{y}$=[$y(0)$, $y(1)$, ..., $y(N-1)$], and the initial samples vector $\boldsymbol{x}_I$ is

$$f_{\boldsymbol{A}|\boldsymbol{Y},\boldsymbol{X}_I}(\boldsymbol{a} \mid \boldsymbol{y},\boldsymbol{x}_I)=\frac{f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{y} \mid \boldsymbol{a},\boldsymbol{x}_I) f_{\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{a},\boldsymbol{x}_I)}{f_{\boldsymbol{Y},\boldsymbol{X}_I}(\boldsymbol{y},\boldsymbol{x}_I)} \tag{8.82}$$

Now consider the variance of the signal $\boldsymbol{y}$ in the argument of the term $f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{y} \mid \boldsymbol{a},\boldsymbol{x}_I)$ in Equation (8.82). The innovation of $y(m)$ can be defined as

$$\begin{aligned}\varepsilon(m) &= y(m)-\sum_{k=1}^{P} a_k\, y(m-k) \\ &= e(m)+n(m)-\sum_{k=1}^{P} a_k\, n(m-k)\end{aligned} \tag{8.83}$$

The variance of $y(m)$, given the previous P samples and the coefficient vector $\boldsymbol{a}$, is the variance of the innovation signal $\varepsilon(m)$, given by

$$\mathrm{Var}[\,y(m)\,|\,y(m-1),\ldots,y(m-P),\boldsymbol{a}\,]=\sigma_\varepsilon^2+\sigma_e^2+\sigma_n^2-\sigma_n^2\sum_{k=1}^{P} a_k^2 \tag{8.84}$$

where σ_e^2 and σ_n^2 are the variance of the excitation signal and the noise respectively. From Equation (8.84), the variance of $y(m)$ is a function of the coefficient vector $\boldsymbol{a}$. Consequently, maximisation of $f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}_I}(\boldsymbol{y}|\boldsymbol{a},\boldsymbol{x}_I)$ with respect to the vector $\boldsymbol{a}$ is a non-linear and non-trivial exercise.

Lim and Oppenheim proposed the following iterative process in which 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

$$f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{Y}}(\boldsymbol{x}|\hat{\boldsymbol{a}},\boldsymbol{y}) = \frac{f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}}(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x})\, f_{\boldsymbol{X}|\boldsymbol{A}}(\boldsymbol{x}|\hat{\boldsymbol{a}})}{f_{\boldsymbol{Y}|\boldsymbol{A}}(\boldsymbol{y}|\hat{\boldsymbol{a}})} \tag{8.85}$$

Consider the likelihood term $f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}}(\boldsymbol{y}|\hat{\boldsymbol{a}},\boldsymbol{x})$. Since the noise is additive, we have

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}}(\boldsymbol{y}\,|\,\hat{\boldsymbol{a}},\boldsymbol{x}) &= f_{\boldsymbol{N}}(\boldsymbol{y}-\boldsymbol{x}) \\ &= \frac{1}{\left(2\pi\sigma_n^2\right)^{N/2}} \exp\left[-\frac{1}{2\sigma_n^2}(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x})\right] \end{aligned} \tag{8.86}$$

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

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{A},\boldsymbol{X}}(\boldsymbol{x}\,|\,\hat{\boldsymbol{a}}) &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}\right) \\ &= \frac{1}{\left(2\pi\sigma_e^2\right)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}\right) \end{aligned} \tag{8.87}$$

where $\boldsymbol{e} = \hat{\boldsymbol{A}}\boldsymbol{x}$ as in Equation (8.69). Substitution of Equations (8.86) and (8.87) in Equation (8.85) yields

$$f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{Y}}(\boldsymbol{x}\,|\,\hat{\boldsymbol{a}},\boldsymbol{y}) = \frac{1}{f_{\boldsymbol{Y}|\boldsymbol{A}}(\boldsymbol{y}\,|\,\hat{\boldsymbol{a}})} \frac{1}{(2\pi\sigma_n\sigma_e)^N} \exp\left[-\frac{1}{2\sigma_n^2}(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x}) - \frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}\right] \tag{8.88}$$

In Equation (8.88), for a given signal $\boldsymbol{y}$ and coefficient vector $\hat{\boldsymbol{a}}$, $f_{\boldsymbol{Y}|\boldsymbol{A}}(\boldsymbol{y}|\hat{\boldsymbol{a}})$ is a constant. From Equation (8.88), the ML signal estimate is obtained by maximising the log-likelihood function as

$$\frac{\partial}{\partial \boldsymbol{a}}\left(\ln f_{\boldsymbol{X}|\boldsymbol{A},\boldsymbol{Y}}\left(\boldsymbol{x}\,|\,\hat{\boldsymbol{a}},\boldsymbol{y}\right)\right)=\frac{\partial}{\partial \boldsymbol{x}}\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}\boldsymbol{x}-\frac{1}{2\sigma_n^2}(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x})\right)=\boldsymbol{0} \tag{8.89}$$

which gives

$$\hat{\boldsymbol{x}}=\sigma_e^2\left(\sigma_n^2\hat{\boldsymbol{A}}^{\mathrm{T}}\hat{\boldsymbol{A}}+\sigma_e^2\mathbf{I}\right)^{-1}\boldsymbol{y} \tag{8.90}$$

The signal estimate of Equation (8.90) 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

$$\hat{\boldsymbol{a}}(\hat{\boldsymbol{x}})=\left(\hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{X}}\right)^{-1}\left(\hat{\boldsymbol{X}}^{\mathrm{T}}\hat{\boldsymbol{x}}\right) \tag{8.91}$$

Equations (8.90) and (8.91) form the basis for an iterative signal restoration/parameter estimation method.

8.6.1 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 $i = 0, 1, ...$

Step 1 Estimate the predictor parameter vector $\hat{\boldsymbol{a}}_i$:

$$\hat{\boldsymbol{a}}_i(\hat{\boldsymbol{x}}_i)=\left(\hat{\boldsymbol{X}}_i^{\mathrm{T}}\hat{\boldsymbol{X}}_i\right)^{-1}\left(\hat{\boldsymbol{X}}_i^{\mathrm{T}}\hat{\boldsymbol{x}}_i\right) \tag{8.92}$$

Step 2 Calculate an estimate of the model gain G using the Parseval's theorem:

$$\frac{1}{N}\sum_{f=0}^{N-1}\frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}e^{-j2\pi fk/N}\right|^2}=\sum_{m=0}^{N-1}y^2(m)-N\hat{\sigma}_n^2 \tag{8.93}$$

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:

$$\hat{P}_{X_iX_i}(f)=\frac{\hat{G}^2}{\left|1-\sum_{k=1}^{P}\hat{a}_{k,i}e^{-j2\pi fk/N}\right|^2} \tag{8.94}$$

Step 4 Calculate the Wiener filter frequency response:

$$\hat{W}_i(f)=\frac{\hat{P}_{X_iX_i}(f)}{\hat{P}_{X_iX_i}(f)+\hat{P}_{N_iN_i}(f)} \tag{8.95}$$

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

$$\hat{X}_{i+1}(f)=\hat{W}_i(f)Y(f) \tag{8.96}$$

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 Goto step 1 and repeat until convergence, or for a specified number of iterations.

Figure 8.13 illustrates a block diagram configuration of a Wiener filter using a linear prediction estimate of the signal spectrum. Figure 8.14 illustrates the result of an iterative restoration of the spectrum of a noisy speech signal.

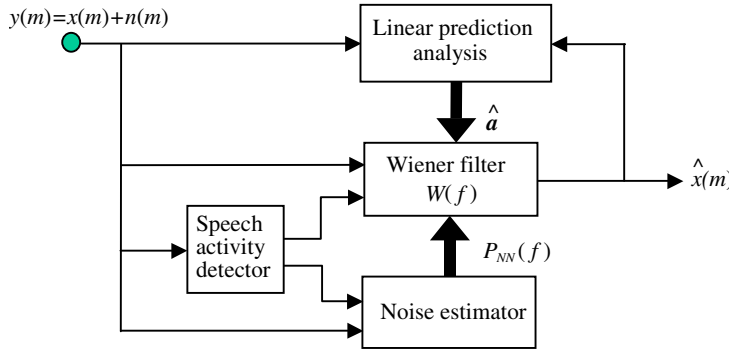


Figure 8.13 Iterative signal restoration based on linear prediction model of speech.

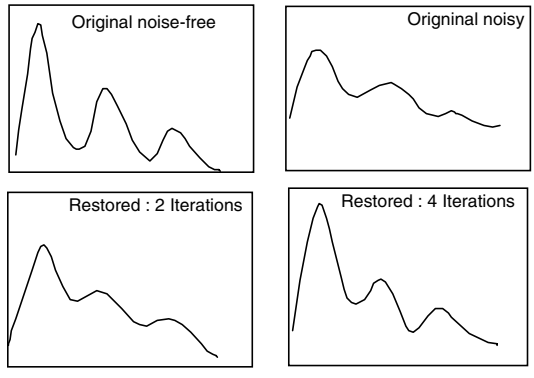


Figure 8.14 Illustration of restoration of a noisy signal with iterative linear prediction based method.

8.6.2 Implementation of Sub-Band Linear Prediction Wiener Filters

Assuming that the noise is additive, the noisy signal in each sub-band is modelled as

$$y_k(m) = x_k(m) + n_k(m) \tag{8.97}$$

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

$$W_k(f) = \frac{P_{X,k}(f)}{P_{Y,k}(f)} = \frac{g_{X,k}^2}{\left|A_{X,k}(f)\right|^2} \frac{\left|A_{Y,k}(f)\right|^2}{g_{Y,k}^2} \tag{8.98}$$

where $P_{X,k}(f)$ and $P_{Y,k}(f)$ are the power spectra of the clean signal and the noisy signal for the k^{th} subband respectively. From Equation (8.98) the square-root Wiener filter is given by

$$W_k^{1/2}(f) = \frac{g_{X,k}}{\left|A_{X,k}(f)\right|} \frac{\left|A_{Y,k}(f)\right|}{g_{Y,k}} \tag{8.99}$$

The linear prediction Wiener filter of Equation (8.99) 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 8.15):

$$z_k(m) = \sum_{i=1}^{P} a_{Xk}(i) z_k(m-i) + \frac{g_X}{g_Y} y_k(m) \tag{8.100}$$

$$\hat{x}_k(m) = \sum_{i=0}^{P} a_{Yk}(i) z_k(m-i) \tag{8.101}$$

where $\hat{x}_k(m)$ is the restored estimate of $x_k(m)$ the clean speech signal and $z_k(m)$ is an intermediate signal.

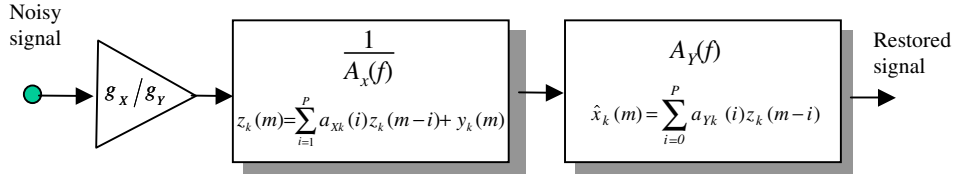


Figure 8.15 A cascade implementation of the LP squared-root Wiener filter.

8.7 Summary

Linear prediction models are used in a wide range of signal processing applications from low-bit-rate speech coding to model-based spectral analysis. We began this chapter with an introduction to linear prediction theory, and considered different methods of formulation of the prediction problem and derivations of the predictor coefficients. The main attraction of the linear prediction method is the closed-form solution of the predictor coefficients, and the availability of a number of efficient and relatively robust methods for solving the prediction equation such as the Levinson–Durbin method. In Section 8.2, we considered the forward, backward and lattice predictors. Although the direct-form implementation of the linear predictor is the most convenient method, for many applications, such as transmission of the predictor coefficients in speech coding, it is advantageous to use the lattice form of the predictor. This is because the lattice form can be conveniently checked for stability, and furthermore a perturbation of the parameter of any section of the lattice structure has a limited and more localised effect. In Section 8.3, we considered a modified form of linear prediction that models the short-term and long-term correlations of the signal. This method can be used for the modelling of signals with a 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. In Section 8.5, the sub-band linear prediction method was formulated. Finally in Section 8.6, a linear prediction model was applied to the restoration of a signal observed in additive noise.

Bibliography

AKAIKE H. (1970) Statistical Predictor Identification, Annals of the Institute of Statistical Mathematics. **22**, pp. 203–217.

AKAIKE H. (1974) A New Look at Statistical Model Identification, IEEE Trans. on Automatic Control, **AC-19**, pp. 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, California.

BURG J.P. (1975) Maximum Entropy Spectral Analysis. P.h.D. thesis, Stanford University, Stanford, California.

COHEN J. and Cohen P. (1975) Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. Halsted, New York.

DRAPER N.R. and Smith H. (1981) Applied Regression Analysis, 2nd Ed. Wiley, New York.

DURBIN J. (1959) Efficient Estimation of Parameters in Moving Average Models. Biometrica, **46**, pp. 306–317.

DURBIN J. (1960) The Fitting of Time Series Models. Rev. Int. Stat. Inst., **28**, pp. 233–244.

FULLER W.A. (1976) Introduction to Statistical Time Series. Wiley, New York.

HANSEN J.H. and CLEMENTS M.A. (1987). Iterative Speech Enhancement with Spectral Constrains. IEEE Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP-87, **1**, pp. 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**, pp. 561–564, New York, April.

HOCKING R.R. (1996): The Analysis of Linear Models. Wiley.

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, pp. 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**, pp. 197-210, June.

LIM J.S. and OPPENHEIM A.V. (1979) Enhancement and Bandwidth Compression of Noisy Speech, Proc. IEEE, **67**, pp. 1586-1604.

MAKOUL J.(1975) Linear Prediction: A Tutorial review. Proceedings of the IEEE, **63**, pp. 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.

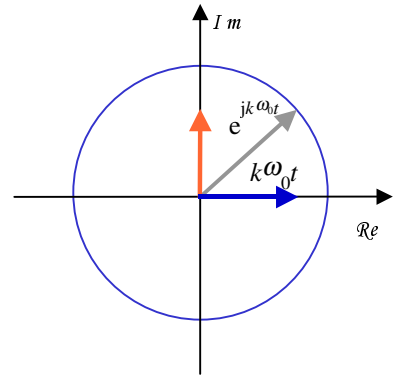
TONG H. (1975) Autoregressive Model Fitting with Noisy Data by Akaike's Information Criterion. IEEE Trans. Information Theory, **IT-23**, pp. 409–48.

STOCKHAM T.G., CANNON T.M. and INGEBRETSEN R.B. (1975) Blind Deconvolution Through Digital Signal Processing. IEEE Proc. **63**, **4**, pp. 678–692.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

9



POWER SPECTRUM AND CORRELATION

The power spectrum reveals the existence, or the absence, of repetitive patterns and correlation structures in a signal process. These structural patterns are important in a wide range of applications such as data forecasting, signal coding, signal detection, radar, pattern recognition, and decision-making systems. The most common method of spectral estimation is based on the fast Fourier transform (FFT). For many applications, FFT-based methods produce sufficiently good results. However, more advanced methods of spectral estimation can offer better frequency resolution, and less variance. This chapter begins with an introduction to the Fourier series and transform and the basic principles of spectral estimation. The classical methods for power spectrum estimation are based on periodograms. Various methods of averaging periodograms, and their effects on the variance of spectral estimates, are considered. We then study the maximum entropy and the model-based spectral estimation methods. We also consider several high-resolution spectral estimation methods, based on eigen-analysis, for the estimation of sinusoids observed in additive white noise.

9.1 Power Spectrum and Correlation

The power spectrum of a signal gives the distribution of the signal power among various frequencies. The power spectrum is the Fourier transform of the correlation function, and reveals information on the correlation structure of the signal. The strength of the Fourier transform in signal analysis and pattern recognition is its ability to reveal spectral structures that may be used to characterise a signal. This is illustrated in Figure 9.1 for the two extreme cases of a sine wave and a purely random signal. For a periodic signal, the power is concentrated in extremely narrow bands of frequencies, indicating the existence of structure and the predictable character of the signal. In the case of a pure sine wave as shown in Figure 9.1(a) the signal power is concentrated in one frequency. For a purely random signal as shown in Figure 9.1(b) the signal power is spread equally in the frequency domain, indicating the lack of structure in the signal.

In general, the more correlated or predictable a signal, 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.

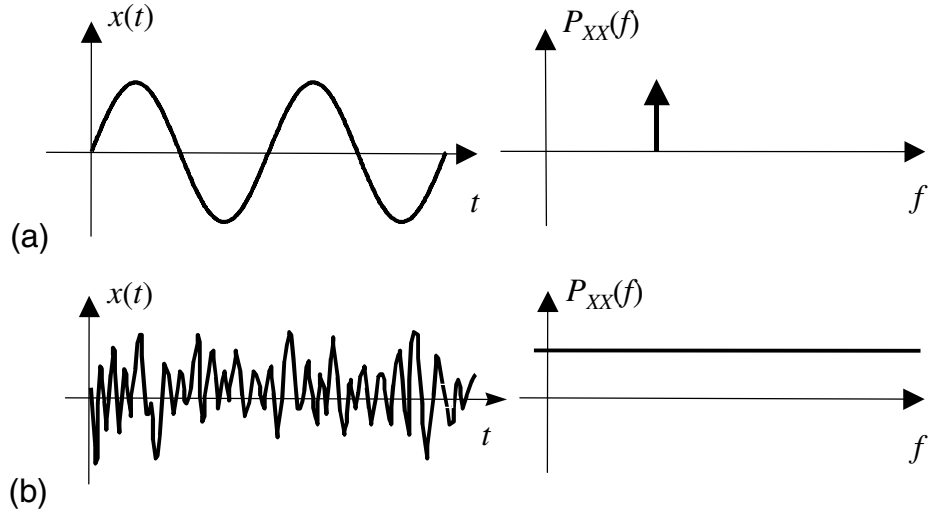


Figure 9.1 The concentration/spread of power in frequency indicates the correlated or random character of a signal: (a) a predictable signal, (b) a random signal.

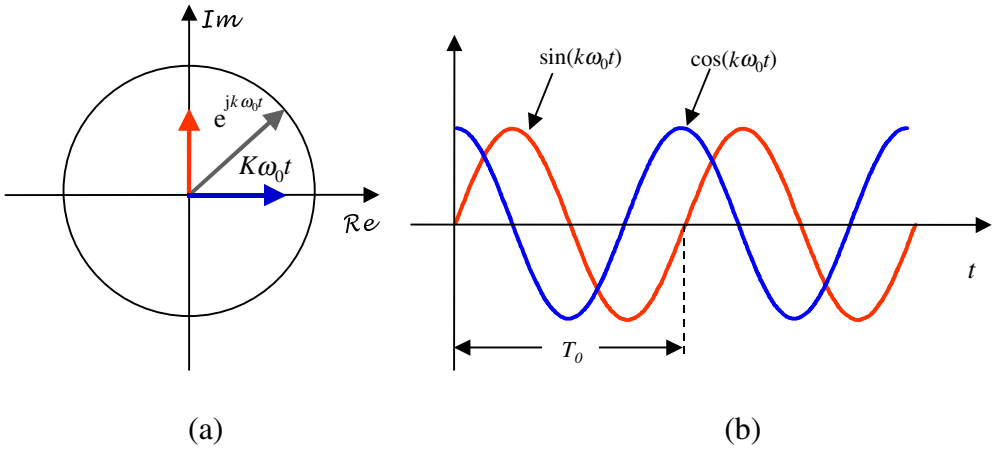


Figure 9.2 Fourier basis functions: (a) real and imaginary parts of a complex sinusoid, (b) vector representation of a complex exponential.

9.2 Fourier Series: Representation of Periodic Signals

The following three sinusoidal functions form the *basis functions* for the Fourier analysis:

$$x_1(t) = \cos \omega_0 t \tag{9.1}$$

$$x_2(t) = \sin \omega_0 t \tag{9.2}$$

$$x_3(t) = \cos \omega_0 t + j \sin \omega_0 t = e^{j\omega_0 t} \tag{9.3}$$

Figure 9.2(a) shows the cosine and the sine components of the complex exponential (cisoidal) signal of Equation (9.3), and Figure 9.2(b) shows a vector representation of the complex exponential in a complex plane with real (Re) and imaginary (Im) dimensions. The Fourier basis functions are periodic with an angular frequency of ω_0 (rad/s) and a period of $T_0 = 2\pi/\omega_0 = 1/F_0$, where F_0 is the frequency (Hz). The following properties make the sinusoids the ideal choice as the elementary building block basis functions for signal analysis and synthesis:

(i) Orthogonality: two sinusoidal functions of *different* frequencies have the following orthogonal property:

$$\int_{-\infty}^{\infty} \sin(\omega_1 t)\sin(\omega_2 t)\,dt = \frac{1}{2}\int_{-\infty}^{\infty} \cos(\omega_1 + \omega_2)\,dt + \frac{1}{2}\int_{-\infty}^{\infty} \cos(\omega_1 - \omega_2)\,dt = 0 \tag{9.4}$$

For harmonically related sinusoids, the integration can be taken over one period. 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 bass, without affecting other frequencies, and in sub-band coding different frequency bands are coded independently and allocated different numbers of bits.

(ii) Sinusoidal functions are infinitely differentiable. This is important, as most signal analysis, synthesis and manipulation methods require the signals to be differentiable.

(iii) Sine and cosine signals of the same frequency have only a phase difference of $\pi/2$ or equivalently a relative time delay of a quarter of one period i.e. $T_0/4$.

Associated with the complex exponential function $e^{j\omega_0 t}$ is a set of harmonically related complex exponentials of the form

$$[1, e^{\pm j\omega_0 t}, e^{\pm j2\omega_0 t}, e^{\pm j3\omega_0 t}, \ldots] \tag{9.5}$$

The set of exponential signals in Equation (9.5) are periodic with a fundamental frequency $\omega_0 = 2\pi/T_0 = 2\pi F_0$, where T_0 is the period and F_0 is the fundamental frequency. These signals form the set of *basis functions* for the Fourier analysis. Any linear combination of these signals of the form

$$\sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \tag{9.6}$$

is also periodic with a period 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 is given by the following synthesis and analysis equations:

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t} \quad k = \cdots \quad -1,0,1,\ldots \qquad \text{(synthesis equation)} \qquad (9.7)$$

$$c_k = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-jk\omega_0 t}\, dt \quad k = \ldots -1,0,1,\ldots \qquad \text{(analysis equation)} \qquad (9.8)$$

The complex-valued coefficient c_k conveys the amplitude (a measure of the strength) and the phase of the frequency content of the signal at $k\omega_0$ (Hz). Note from Equation (9.8) that the coefficient c_k *may be interpreted as a measure of the correlation of the signal* $x(t)$ *and the complex exponential* $e^{-jk\omega_0 t}$.

9.3 Fourier Transform: Representation of Aperiodic Signals

The Fourier series representation of periodic signals consist of harmonically related spectral lines spaced at integer multiples of the fundamental frequency. The Fourier representation of aperiodic signals can be developed by regarding an aperiodic signal as a special case of a periodic signal with an infinite period. If the period of a signal is infinite then the signal does not repeat itself, and is aperiodic.

Now consider the discrete spectra of a periodic signal with a period of T_0, as shown in Figure 9.3(a). As the period T_0 is increased, 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 aperiodic), the discrete spectral lines merge and form a continuous spectrum. Therefore the Fourier equations for an aperiodic signal (known as the Fourier transform) must reflect the fact that the frequency spectrum of an aperiodic signal is continuous. Hence, to obtain the Fourier transform relation, the discrete-frequency variables and operations in the Fourier series Equations (9.7) and (9.8) should be replaced by their continuous-frequency counterparts. That is, the discrete summation sign Σ should be replaced by the continuous summation integral $\int$, the discrete harmonics of the fundamental frequency kF_0 should be replaced by the continuous frequency variable f, and the discrete frequency spectrum c_k should be replaced by a continuous frequency spectrum say $X(f)$.

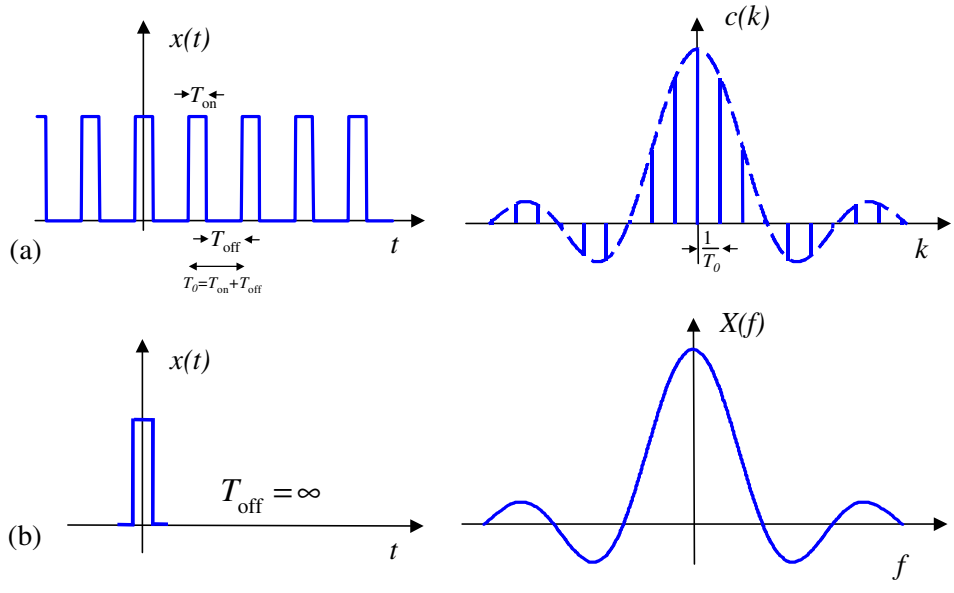


Figure 9.3 (a) A periodic pulse train and its line spectrum. (b) A single pulse from the periodic train in (a) with an imagined "off" duration of infinity; its spectrum is the envelope of the spectrum of the periodic signal in (a).

The Fourier synthesis and analysis equations for aperiodic signals, the so-called *Fourier transform pair*, are given by

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \tag{9.9}$$

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \tag{9.10}$$

Note from Equation (9.10), 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 existence and computability of the Fourier transform integral of a signal $x(t)$ is that the signal must have finite energy:

$$\int_{-\infty}^{\infty} |x(t)|^2 dt < \infty \tag{9.11}$$

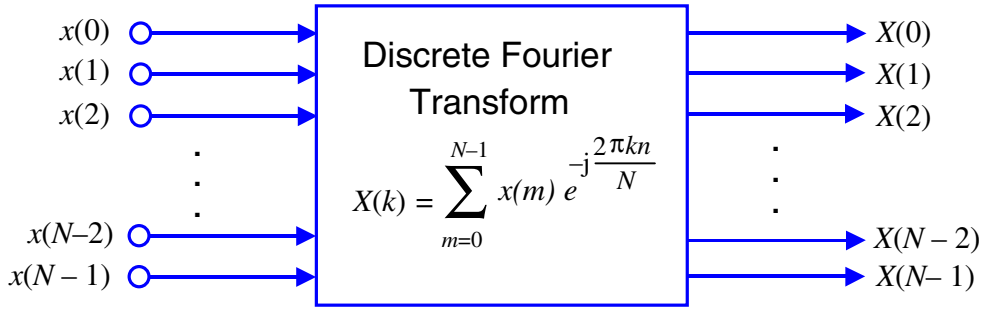


Figure 9.4 Illustration of the DFT as a parallel-input, parallel-output processor.

9.3.1 Discrete Fourier Transform (DFT)

For a finite-duration, discrete-time signal $x(m)$ of length N samples, the discrete Fourier transform (DFT) is defined as N uniformly spaced spectral samples

$$X(k) = \sum_{m=0}^{N-1} x(m) e^{-j(2\pi/N)mk}, \qquad k = 0, \ldots, N-1 \tag{9.12}$$

(see Figure9.4). The inverse discrete Fourier transform (IDFT) is given by

$$x(m) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(2\pi/N)mk}, \quad m = 0, \ldots, N-1 \tag{9.13}$$

From Equation (9.13), the direct calculation of the Fourier transform requires $N(N-1)$ multiplications and a similar number of additions. Algorithms that reduce the computational complexity of the discrete Fourier transform are known as fast Fourier transforms (FFT) methods. FFT methods utilise the periodic and symmetric properties of $e^{-j2\pi/N}$ to avoid redundant calculations.

9.3.2 Time/Frequency Resolutions, The Uncertainty Principle

Signals such as speech, music or image are composed of non-stationary (i.e. time-varying and/or space-varying) events. For example, speech is composed of a string of short-duration sounds called phonemes, and an

image is composed of various objects. When using the DFT, it is desirable to have high enough time and space resolution in order to obtain the spectral characteristics of each individual elementary event or object in the input signal. However, there is a fundamental trade-off between the length, i.e. the time or space resolution, of the input signal and the frequency resolution of the output spectrum. The DFT takes as the input a window of N uniformly spaced time-domain samples [$x(0)$, $x(1)$, …, $x(N-1)$] of duration $\Delta T=N.T_s$, and outputs N spectral samples [$X(0)$, $X(1)$, …, $X(N-1)$] spaced uniformly between zero Hz and the sampling frequency $F_s=1/T_s$ Hz. Hence the frequency resolution of the DFT spectrum Δf, i.e. the space between successive frequency samples, is given by

$$\Delta f = \frac{1}{\Delta T} = \frac{1}{NT_s} = \frac{F_s}{N} \tag{9.14}$$

Note that the frequency resolution Δf and the time resolution ΔT are inversely proportional in that they cannot both be simultanously increased; in fact, $\Delta T \Delta f=1$. This is known as the uncertainty principle.

9.3.3 Energy-Spectral Density and Power-Spectral Density

Energy, or power, spectrum analysis is concerned with the distribution of the signal energy or power in the frequency domain. For a deterministic discrete-time signal, the energy-spectral density is defined as

$$\left|X(f)\right|^2 = \left|\sum_{m=-\infty}^{\infty} x(m)e^{-j2\pi fm}\right|^2 \tag{9.15}$$

The energy spectrum of $x(m)$ may be expressed as the Fourier transform of the autocorrelation function of $x(m)$:

$$\begin{aligned} \left|X(f)\right|^2 &= X(f)X^*(f) \\ &= \sum_{m=-\infty}^{\infty} r_{xx}(m)e^{-j2\pi fm} \end{aligned} \tag{9.16}$$

where the variable $r_{xx}(m)$ is the autocorrelation function of $x(m)$. 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:

$$P_{XX}(f) = \sum_{m=-\infty}^{\infty} r_{xx}(m) e^{-j2\pi fm} \tag{9.17}$$

where the autocorrelation function $r_{xx}(m)$ is defined as

$$r_{xx}(m) = \mathcal{E}[x(m)x(m+k)] \tag{9.18}$$

In practice, the autocorrelation function is estimated from a signal record of length N samples as

$$\hat{r}_{xx}(m) = \frac{1}{N - |m|} \sum_{k=0}^{N-|m|-1} x(k)x(k+m), \quad k = 0, \ldots, N-1 \tag{9.19}$$

In Equation (9.19), 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

$$w(m) = \begin{cases} 1 - \dfrac{|m|}{N}, & |m| \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{9.20}$$

Multiplication of Equation (9.19) by the window of Equation (9.20) yields

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-|m|-1} x(k)x(k+m) \tag{9.21}$$

The expectation of the windowed correlation estimate $\hat{r}_{xx}(m)$ is given by

$$\begin{aligned}\mathcal{E}\left[\hat{r}_{xx}(m)\right] &= \frac{1}{N}\sum_{k=0}^{N-|m|-1}\mathcal{E}\left[x(k)x(k+m)\right] \\ &= \left(1-\frac{|m|}{N}\right)r_{xx}(m)\end{aligned} \tag{9.22}$$

In Jenkins and Watts, it is shown that the variance of $\hat{r}_{xx}(m)$ is given by

$$\mathrm{Var}[\hat{r}_{xx}(m)] \approx \frac{1}{N}\sum_{k=-\infty}^{\infty}\left[r_{xx}^2(k) + r_{xx}(k-m)r_{xx}(k+m)\right] \tag{9.23}$$

From Equations (9.22) and (9.23), $\hat{r}_{xx}(m)$ is an asymptotically unbiased and consistent estimate.

9.4 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 1899. The periodogram is defined as

$$\begin{aligned}\hat{P}_{XX}(f) &= \frac{1}{N}\left|\sum_{m=0}^{N-1}x(m)e^{-j2\pi fm}\right|^2 \\ &= \frac{1}{N}|X(f)|^2\end{aligned} \tag{9.24}$$

The power-spectral density function, or power spectrum for short, defined in Equation (9.24), 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.

9.4.1 The Mean and Variance of Periodograms

The mean of the periodogram is obtained by taking the expectation of Equation (9.24):

$$\begin{aligned}\mathcal{E}[\hat{P}_{XX}(f)] &= \frac{1}{N}\mathcal{E}\left[\left|X(f)\right|^2\right] \\ &= \frac{1}{N}\mathcal{E}\left[\sum_{m=0}^{N-1} x(m)e^{-j2\pi fm}\sum_{n=0}^{N-1} x(n)e^{j2\pi fn}\right] \\ &= \sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)r_{xx}(m)e^{-j2\pi fm}\end{aligned} \tag{9.25}$$

As the number of signal samples N increases, we have

$$\lim_{N\to\infty}\mathcal{E}[\hat{P}_{XX}(f)] = \sum_{m=-\infty}^{\infty} r_{xx}(m)e^{-j2\pi fm} = P_{XX}(f) \tag{9.26}$$

For a Gaussian random sequence, the variance of the periodogram can be obtained as

$$\mathrm{Var}[\hat{P}_{XX}(f)] = P_{XX}^2(f)\left[1+\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{9.27}$$

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.

9.4.2 Averaging Periodograms (Bartlett Method)

In this method, several periodograms, from different segments of a signal, are averaged in order to reduce the variance of the periodogram. The Bartlett periodogram is obtained as the average of K periodograms as

$$\hat{P}_{XX}^{B}(f) = \frac{1}{K}\sum_{i=1}^{K}\hat{P}_{XX}^{(i)}(f) \tag{9.28}$$

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

$$\begin{aligned}\mathcal{E}[\hat{P}_{XX}^{B}(f)] &= \mathcal{E}[\hat{P}_{XX}^{(i)}(f)] \\ &= \sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)r_{xx}(m)e^{-j2\pi fm} \\ &= \frac{1}{N}\int_{-1/2}^{1/2} P_{XX}(\nu)\left[\frac{\sin\pi(f-\nu)N}{\sin\pi(f-\nu)}\right]^2 d\nu\end{aligned} \tag{9.29}$$

where $(\sin\pi fN/\sin\pi f)^2/N$ is the frequency response of the triangular window $1-|m|/N$. From Equation (9.29), 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

$$\mathrm{Var}\left[\hat{P}_{XX}^{B}(f)\right] = \frac{1}{K}P_{XX}^{2}(f)\left[1+\left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{9.30}$$

9.4.3 Welch Method: Averaging Periodograms from Overlapped and Windowed Segments

In this method, a signal $x(m)$, of length M samples, is divided into K overlapping segments of length N, and each segment is windowed prior to computing the periodogram. The i^{th} segment is defined as

$$x_i(m) = x(m+iD), \qquad m=0,\ldots,N-1,\ i=0,\ldots,K-1 \tag{9.31}$$

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

$$\hat{P}_{XX}^{(i)}(f)]=\frac{1}{NU}\left|\sum_{m=0}^{N-1}w(m)x_i(m)e^{-j2\pi fm}\right|^2 \tag{9.32}$$

where $w(m)$ is the window function and U is the power in the window function, given by

$$U=\frac{1}{N}\sum_{m=0}^{N-1}w^2(m) \tag{9.33}$$

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:

$$\hat{P}_{XX}^{W}(f)=\frac{1}{K}\sum_{i=0}^{K-1}\hat{P}_{XX}^{(i)}(f) \tag{9.34}$$

Using Equations (9.32) and (9.34), the expectation of $\hat{P}_{XX}^{W}(f)$ can be obtained as

$$\begin{aligned}\mathcal{E}[P_{XX}^{W}(f)]&=\mathcal{E}[\hat{P}_{XX}^{(i)}(f)]\\&=\frac{1}{NU}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}w(n)w(m)\mathcal{E}[x_i(m)x_i(n)]e^{-j2\pi f(n-m)}\\&=\frac{1}{NU}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}w(n)w(m)r_{xx}(n-m)e^{-j2\pi f(n-m)}\\&=\int_{-1/2}^{1/2}P_{XX}(\nu)W(\nu-f)d\nu\end{aligned} \tag{9.35}$$

where

$$W(f)=\frac{1}{NU}\left|\sum_{m=0}^{N-1}w(m)e^{-j2\pi fm}\right|^2 \tag{9.36}$$

and the variance of the Welch estimate is given by

$$\mathrm{Var}[\hat{P}_{XX}^{W}(f)]=\frac{1}{K^2}\sum_{i=0}^{K-1}\sum_{j=0}^{K-1}\mathcal{E}\left[\hat{P}_{XX}^{(i)}(f)\hat{P}_{XX}^{(j)}(f)\right]-\left(\mathcal{E}\left[\hat{P}_{XX}^{W}(f)\right]\right)^2 \tag{9.37}$$

Welch has shown that for the case when there is no overlap, $D=N$,

$$\mathrm{Var}[P_{XX}^{W}(f)]=\frac{\mathrm{Var}[P_{XX}^{(i)}(f)]}{K_1}\approx\frac{P_{XX}^{2}(f)}{K_1} \tag{9.38}$$

and for half-overlap, $D=N/2$,

$$\mathrm{Var}[\hat{P}_{XX}^{W}(f)]=\frac{9}{8K_2}P_{XX}^{2}(f)] \tag{9.39}$$

9.4.4 Blackman–Tukey Method

In this method, an estimate of a signal power spectrum is obtained from the Fourier transform of the windowed estimate of the autocorrelation function as

$$\hat{P}_{XX}^{BT}(f)=\sum_{m=-(N-1)}^{N-1}w(m)\hat{r}_{xx}(m)e^{-j2\pi fm} \tag{9.40}$$

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

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)]=\sum_{m=-(N-1)}^{N-1}\mathcal{E}[\hat{r}_{xx}(m)]w(m)\,e^{-j2\pi fm} \tag{9.41}$$

Now $\mathcal{E}[\hat{r}_{xx}(m)]=r_{xx}(m)w_B(m)$, where $w_B(m)$ is the Bartlett, or triangular, window. Equation (9.41) may be written as

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)] = \sum_{m=-(N-1)}^{N-1} r_{xx}(m) w_c(m) e^{-j2\pi fm} \tag{9.42}$$

where $w_c(m) = w_B(m)w(m)$. The right-hand side of Equation (9.42) can be written in terms of the Fourier transform of the autocorrelation and the window functions as

$$\mathcal{E}[\hat{P}_{XX}^{BT}(f)] = \int_{-1/2}^{1/2} P_{XX}(\nu) W_c(f-\nu) d\nu \tag{9.43}$$

where $W_c(f)$ is the Fourier transform of $w_c(m)$. The variance of the Blackman–Tukey estimate is given by

$$\text{Var}[\hat{P}_{XX}^{BT}(f)] \approx \frac{U}{N} P_{XX}^2(f) \tag{9.44}$$

where U is the energy of the window $w_c(m)$.

9.4.5 Power Spectrum Estimation from Autocorrelation of Overlapped Segments

In the Blackman–Tukey method, in calculating a correlation sequence of length N from a signal record of length N, progressively fewer samples are admitted in estimation of $\hat{r}_{xx}(m)$ as the lag m approaches the signal length N. Hence the variance of $\hat{r}_{xx}(m)$ increases with the lag m. This problem can be solved by using a signal of length $2N$ samples for calculation of N correlation values. In a 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

$$x_i(m) = x(m + iD), \quad \begin{matrix} m = 0, 1, \ldots, 2N-1 \\ i = 0, 1, \ldots, K-1 \end{matrix} \tag{9.45}$$

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

$$\hat{r}_{xx}(m) = \frac{1}{N} \sum_{k=0}^{N-1} x_i(k) x_i(k+m), \quad m = 0, 1, \ldots, N-1 \tag{9.46}$$

In Equation (9.46), the estimate of each correlation value is obtained as the averaged sum of N products.

9.5 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

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + \sum_{k=0}^{Q} b_k e(m-k) \tag{9.47}$$

where $x(m)$ is the model output, $e(m)$ is the input, and the a_k and b_k are the parameters of the model. Equation (9.47) is known as an auto-regressive-moving-average (ARMA) model. The system function $H(z)$ of the discrete linear time-invariant model of Equation (9.47) is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{Q} b_k z^{-k}}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{9.48}$$

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:

$$P_{XX}(f)=P_{EE}(f)|H(f)|^2 \tag{9.49}$$

where $H(f)$ is the frequency response of the model and $P_{EE}(f)$ is the input power spectrum. Assuming that the input is a white noise process with unit variance, i.e. $P_{EE}(f)=1$, Equation (9.49) becomes

$$P_{XX}(f)=|H(f)|^2 \tag{9.50}$$

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.

9.5.1 Maximum–Entropy Spectral Estimation

The power spectrum of a stationary signal is defined as the Fourier transform of the autocorrelation sequence:

$$P_{XX}(f) = \sum_{n=-\infty}^{\infty} r_{xx}(m)e^{-j2\pi fm} \tag{9.51}$$

Equation (9.51) 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 defined as

$$H\left[P_{XX}(f)\right]=\int_{-1/2}^{1/2}\ln P_{XX}(f)\,df \tag{9.52}$$

To obtain the maximum-entropy correlation estimate, we differentiate Equation (9.53) with respect to the unknown values of the correlation coefficients, and set the derivative to zero:

$$\frac{\partial H[P_{XX}(f)]}{\partial r_{xx}(m)}=\int_{-1/2}^{1/2}\frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)}\,df=0 \quad \text{for } |m|>P \tag{9.53}$$

Now, from Equation (9.17), the derivative of the power spectrum with respect to the autocorrelation values is given by

$$\frac{\partial P_{XX}(f)}{\partial r_{xx}(m)}=e^{-j2\pi fm} \tag{9.54}$$

From Equation (9.51), for the derivative of the logarithm of the power spectrum, we have

$$\frac{\partial \ln P_{XX}(f)}{\partial r_{xx}(m)}=P_{XX}^{-1}(f)e^{-j2\pi fm} \tag{9.55}$$

Substitution of Equation (9.55) in Equation (9.53) gives

$$\int_{-1/2}^{1/2}P_{XX}^{-1}(f)e^{-j2\pi fm}\,df=0 \qquad \text{for } |m|>P \tag{9.56}$$

Assuming that $P_{XX}^{-1}(f)$ is integrable, it may be associated with an autocorrelation sequence $c(m)$ as

$$P_{XX}^{-1}(f)=\sum_{m=-\infty}^{\infty}c(m)\,e^{-j2\pi fm} \tag{9.57}$$

where

$$c(m)=\int_{-1/2}^{1/2}P_{XX}^{-1}(f)e^{j2\pi fm}\,df \tag{9.58}$$

From Equations (9.56) and (9.58), we have $c(m)=0$ for $|m| > P$. Hence, from Equation (9.57), the inverse of the maximum-entropy power spectrum may be obtained from the Fourier transform of a finite-length autocorrelation sequence as

$$P_{XX}^{-1}(f) = \sum_{m=-P}^{P} c(m) e^{-j2\pi fm} \tag{9.59}$$

and the maximum-entropy power spectrum is given by

$$\hat{P}_{XX}^{ME}(f) = \frac{1}{\sum_{m=-P}^{P} c(m) e^{-j2\pi fm}} \tag{9.60}$$

Since the denominator polynomial in Equation (9.60) 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

$$\sum_{m=-P}^{P} c(m) z^{-m} = \frac{1}{\sigma^2} A(z) A(z^{-1}) \tag{9.61}$$

where $1/\sigma^2$ is a gain term, and $A(z)$ is a polynomial of order P defined as

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-P} \tag{9.62}$$

From Equations (9.60) and (9.61), the maximum-entropy power spectrum may be expressed as

$$\hat{P}_{XX}^{ME}(f) = \frac{\sigma^2}{A(z)A(z^{-1})} \tag{9.63}$$

Equation (9.63) shows that the maximum-entropy power spectrum estimate is the power spectrum of an autoregressive (AR) model. Equation (9.63) was obtained by maximising the entropy of the power spectrum with respect to the unknown autocorrelation values. The known values of the autocorrelation function can be used to obtain the coefficients of the AR model of Equation (9.63), as discussed in the next section.

9.5.2 Autoregressive Power Spectrum Estimation

In the preceding section, it was shown that the maximum-entropy spectrum is equivalent to the spectrum of an autoregressive model of the signal. An autoregressive, or linear prediction model, described in detail in Chapter 8, is defined as

$$x(m)=\sum_{k=1}^{P} a_k x(m-k)+e(m) \tag{9.64}$$

where $e(m)$ is a random signal of variance σ_e^2. The power spectrum of an autoregressive process is given by

$$P_{XX}^{AR}(f)=\frac{\sigma_e^2}{\left|1-\sum_{k=1}^{P} a_k e^{-j2\pi fk}\right|^2} \tag{9.65}$$

An AR model extrapolates the correlation sequence beyond the range for which estimates are available. The relation between the autocorrelation values and the AR model parameters is obtained by multiplying both sides of Equation (9.64) by $x(m$-j) and taking the expectation:

$$\mathcal{E}[x(m)x(m-j)] = \sum_{k=1}^{P} a_k \mathcal{E}[x(m-k)x(m-j)] + \mathcal{E}[e(m)x(m-j)] \tag{9.66}$$

Now for the optimal model coefficients the random input $e(m)$ is orthogonal to the past samples, and Equation (9.66) becomes

$$r_{xx}(j) = \sum_{k=1}^{P} a_k r_{xx}(j-k), \qquad j=1, 2, \ldots \tag{9.67}$$

Given P+1 correlation values, Equation (9.67) can be solved to obtain the AR coefficients a_k. Equation (9.67) can also be used to extrapolate the correlation sequence. The methods of solving the AR model coefficients are discussed in Chapter 8.

9.5.3 Moving-Average Power Spectrum Estimation

A moving-average model is also known as an all-zero or a finite impulse response (FIR) filter. A signal $x(m)$, modelled as a moving-average process, is described as

$$x(m)=\sum_{k=0}^{Q} b_k\, e(m-k) \tag{9.68}$$

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

$$\begin{aligned} r_{xe}(m) &= \mathcal{E}[x(j)e(j-m)] \\ &= \mathcal{E}\left[\sum_{k=0}^{Q} b_k e(j-k)e(j-m)\right] = \sigma_e^2 b_m \end{aligned} \tag{9.69}$$

and the autocorrelation function of a moving average process is

$$r_{xx}(m)=\begin{cases} \sigma_e^2 \displaystyle\sum_{k=0}^{Q-|m|} b_k b_{k+m}, & |m|\le Q \\ 0, & |m|>Q \end{cases} \tag{9.70}$$

From Equation (9.70), 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

$$P_{XX}^{MA} = \sum_{m=-Q}^{Q} r_{xx}(m)e^{-\mathrm{j}2\pi fm} \tag{9.71}$$

Note that the moving-average spectral estimation is identical to the Blackman–Tukey method of estimating periodograms from the autocorrelation sequence.

9.5.4 Autoregressive Moving-Average Power Spectrum Estimation

The ARMA, or pole–zero, model is described by Equation (9.47). The relationship between the ARMA parameters and the autocorrelation sequence can be obtained by multiplying both sides of Equation (9.47) by $x(m–j)$ and taking the expectation:

$$r_{xx}(j) = -\sum_{k=1}^{P} a_k r_{xx}(j-k) + \sum_{k=0}^{Q} b_k r_{xe}(j-k) \tag{9.72}$$

The moving-average part of Equation (9.72) influences the autocorrelation values only up to the lag of Q. Hence, for the autoregressive part of Equation (9.72), we have

$$r_{xx}(m) = -\sum_{k=1}^{P} a_k r_{xx}(m-k) \quad \text{for} \quad m > Q \tag{9.73}$$

Hence Equation (9.73) can be used to obtain the coefficients a_k, which may then be substituted in Equation (9.72) for solving the coefficients b_k. Once the coefficients of an ARMA model are identified, the spectral estimate is given by

$$P_{XX}^{ARMA}(f) = \sigma_e^2 \frac{\left| \sum_{k=0}^{Q} b_k e^{-j2\pi fk} \right|^2}{\left| 1 + \sum_{k=1}^{P} a_k e^{-j2\pi fk} \right|^2} \tag{9.74}$$

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.

9.6 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:

(a) the signal subspace composed of the *principle* eigenvectors associated with the largest eigenvalues;
(b) 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.

9.6.1 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 9.5. The AR model for a sinusoid of frequency F_i at a sampling rate of F_s is given by

$$x(m)=2\cos(2\pi F_i / F_s)\, x(m-1)-x(m-2)+A\delta(m-t_0) \tag{9.75}$$

where $A\delta(m–t_0)$ is the initial impulse for a sine wave of amplitude A. In general, a signal composed of P real sinusoids can be modelled by an AR model of order $2P$ as

$$x(m)=\sum_{k=1}^{2P} a_k x(m-k)+A\delta(m-t_0) \tag{9.76}$$

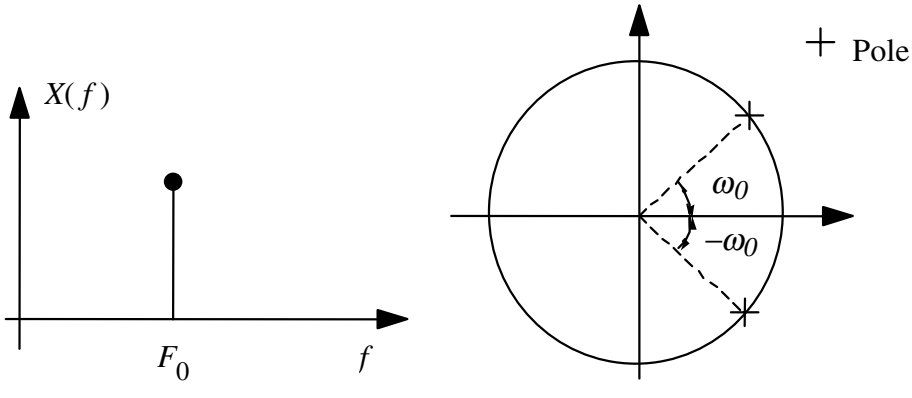


Figure 9.5 A second order all pole model of a sinusoidal signal.

The transfer function of the AR model is given by

$$H(z) = \frac{A}{1 - \sum_{k=1}^{2P} a_k z^{-k}} = \frac{A}{\prod_{k=1}^{P} (1 - e^{-j2\pi F_k} z^{-1})(1 - e^{+j2\pi F_k} z^{-1})} \tag{9.77}$$

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

$$\begin{aligned} y(m) &= x(m) + n(m) \\ &= \sum_{k=1}^{2P} a_k x(m-k) + n(m) \end{aligned} \tag{9.78}$$

Substituting $[y(m–k)–n(m–k)]$ for $x(m–k)$ in Equation (9.73) yields

$$y(m) - \sum_{k=1}^{2P} a_k y(m-k) = n(m) - \sum_{k=1}^{2P} a_k n(m-k) \tag{9.79}$$

From Equation (9.79), the noisy sinusoidal signal $y(m)$ can be modelled by an ARMA process in which the AR and the MA sections are identical, and the input is the noise process. Equation (9.79) can also be expressed in a vector notation as

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{a} = \boldsymbol{n}^{\mathrm{T}}\boldsymbol{a} \tag{9.80}$$

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 (9.80) by the vector $\boldsymbol{y}$ and take the expectation:

$$\mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}]\boldsymbol{a} = \mathcal{E}[\boldsymbol{y}\boldsymbol{n}^{\mathrm{T}}]\boldsymbol{a} \tag{9.81}$$

or

$$\boldsymbol{R}_{\boldsymbol{yy}}\boldsymbol{a} = \boldsymbol{R}_{\boldsymbol{yn}}\boldsymbol{a} \tag{9.82}$$

where $\mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}]=\boldsymbol{R}_{\boldsymbol{yy}}$, and $\mathcal{E}[\boldsymbol{y}\boldsymbol{n}^{\mathrm{T}}]=\boldsymbol{R}_{\boldsymbol{yn}}$ can be written as

$$\begin{aligned}\boldsymbol{R}_{yn} &= \mathcal{E}[(\boldsymbol{x}+\boldsymbol{n})\boldsymbol{n}^{\mathrm{T}}] \\ &= \mathcal{E}[\boldsymbol{n}\boldsymbol{n}^{\mathrm{T}}] = \boldsymbol{R}_{nn} = \sigma_n^2\mathbf{I}\end{aligned} \tag{9.83}$$

where σ_n^2 is the noise variance. Using Equation (9.83), Equation (9.82) becomes

$$\boldsymbol{R}_{yy}\boldsymbol{a} = \sigma_n^2\boldsymbol{a} \tag{9.84}$$

Equation (9.84) is in the form of an eigenequation. 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

$$1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_2 z^{-2P+2} + a_1 z^{-2P+1} + z^{-2P} = 0 \tag{9.85}$$

Note that for sinusoids, the AR parameters form a symmetric polynomial; that is $a_k = a_{2P-k}$. The frequencies F_k of the sinusoids can be obtained from the roots z_k of Equation (9.85) using the relation

$$z_k = e^{j2\pi F_k} \tag{9.86}$$

The powers of the sinusoids are calculated as follows. For P sinusoids observed in additive white noise, the autocorrelation function is given by

$$r_{yy}(k) = \sum_{i=1}^{P} P_i \cos 2k\pi F_i + \sigma_n^2\,\delta(k) \tag{9.87}$$

where $P_i = A_i^2/2$ is the power of the sinusoid $A_i \sin(2\pi F_i)$, and white noise affects only the correlation at lag zero $r_{yy}(0)$. Hence Equation (9.87) for the correlation lags $k = 1, \ldots, P$ can be written as

$$\begin{pmatrix} \cos 2\pi F_1 & \cos 2\pi F_2 & \dots & \cos 2\pi F_P \\ \cos 4\pi F_1 & \cos 4\pi F_2 & \dots & \cos 4\pi F_P \\ \vdots & \vdots & \ddots & \vdots \\ \cos 2P\pi F_1 & \cos 2P\pi F_2 & \dots & \cos 2P\pi F_P \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_P \end{pmatrix} = \begin{pmatrix} r_{yy}(1) \\ r_{yy}(2) \\ \vdots \\ r_{yy}(P) \end{pmatrix} \tag{9.88}$$

Given an estimate of the frequencies F_i from Equations (9.85) and (86), and an estimate of the autocorrelation function $\hat{r}_{yy}(k)$, Equation (9.88) can be solved to obtain the powers of the sinusoids P_i. The noise variance can then be obtained from Equation (9.87) as

$$\sigma_n^2 = r_{yy}(0) - \sum_{i=1}^{P} P_i \tag{9.89}$$

9.6.2 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

$$y(m) = \sum_{k=1}^{P} A_k e^{-\mathrm{j}(2\pi F_k m + \phi_k)} + n(m) \tag{9.90}$$

An N-sample vector $\boldsymbol{y}=[y(m), \ldots, y(m+N-1)]$ of the noisy signal can be written as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} + \boldsymbol{n} \\ &= \boldsymbol{S}\boldsymbol{a} + \boldsymbol{n} \end{aligned} \tag{9.91}$$

where the signal vector $\boldsymbol{x}=\boldsymbol{S}\boldsymbol{a}$ is defined as

$$\begin{pmatrix} x(m) \\ x(m+1) \\ \vdots \\ x(m+N-1) \end{pmatrix} = \begin{pmatrix} e^{\mathrm{j}2\pi F_1 m} & e^{\mathrm{j}2\pi F_2 m} & \dots & e^{\mathrm{j}2\pi F_P m} \\ e^{\mathrm{j}2\pi F_1 (m+1)} & e^{\mathrm{j}2\pi F_2 (m+1)} & \dots & e^{\mathrm{j}2\pi F_P (m+1)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{\mathrm{j}2\pi F_1 (m+N-1)} & e^{\mathrm{j}2\pi F_2 (m+N-1)} & \dots & e^{\mathrm{j}2\pi F_P (m+N-1)} \end{pmatrix} \begin{pmatrix} A_1 e^{\mathrm{j}2\pi\phi_1} \\ A_2 e^{\mathrm{j}2\pi\phi_2} \\ \vdots \\ A_P e^{\mathrm{j}2\pi\phi_P} \end{pmatrix} \tag{9.92}$$

The matrix $\boldsymbol{S}$ and the vector $\boldsymbol{a}$ are defined on the right-hand side of Equation (9.92). 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

$$\begin{aligned} \boldsymbol{R}_{yy} &= \boldsymbol{R}_{xx} + \boldsymbol{R}_{nn} \\ &= \boldsymbol{S}\boldsymbol{P}\boldsymbol{S}^{\mathrm{H}} + \sigma_n^2 \mathbf{I} \end{aligned} \tag{9.93}$$

where $\boldsymbol{R}_{xx}=\boldsymbol{S}\boldsymbol{P}\boldsymbol{S}^{\mathrm{H}}$ 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

$$\boldsymbol{P} = \boldsymbol{a}\boldsymbol{a}^{\mathrm{H}} = \mathrm{diag}[P_1, P_2, \ldots, P_P] \tag{9.94}$$

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

$$\boldsymbol{R}_{xx} = \sum_{k=1}^{P} P_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{H}} \tag{9.95}$$

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 $\boldsymbol{R}_{xx}$

$$\begin{aligned} \boldsymbol{R}_{xx} &= \sum_{k=1}^{N} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \\ &= \sum_{k=1}^{P} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \end{aligned} \tag{9.96}$$

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, $\lambda_{P+1}=\lambda_{P+2}$, ..., $\lambda_N=0$. 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

$$\boldsymbol{R}_{\boldsymbol{nn}} = \sigma_n^2 \mathbf{I} = \sigma_n^2 \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \tag{9.97}$$

The correlation matrix of the noisy signal may be expressed in terms of its eigenvectors and the associated eigenvalues of the noisy signal as

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{yy}} &= \sum_{k=1}^{P} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} + \sigma_n^2 \sum_{k=1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \\ &= \sum_{k=1}^{P} \left(\lambda_k + \sigma_n^2\right) \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} + \sigma_n^2 \sum_{k=P+1}^{N} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{H}} \end{aligned} \tag{9.98}$$

From Equation (9.98), the eigenvectors and the eigenvalues of the correlation matrix of the noisy signal can be partitioned into two disjoint subsets (see Figure 9.6). The set of eigenvectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_P\}$, associated with the P largest eigenvalues span the *signal subspace* and are called the *principal eigenvectors*. 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

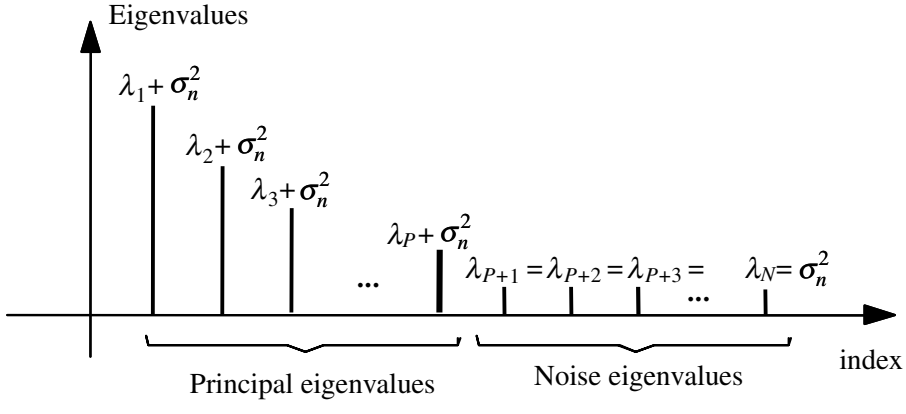


Figure 9.6 Decomposition of the eigenvalues of a noisy signal into the principal eigenvalues and the noise eigenvalues.

$$\boldsymbol{s}_i^{\mathrm{H}}(f)\boldsymbol{v}_k = \sum_{m=0}^{N-1} v_k(m)e^{-j2\pi F_i m} = 0 \qquad i = 1,\ldots,P \quad k = P+1,\ldots,N \tag{9.99}$$

Equation (9.99) 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:

$$\sum_{k=P+1}^{N} \boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k \tag{9.100}$$

In the MUSIC algorithm, the power spectrum estimate is defined as

$$P_{XX}(f) = \sum_{k=P+1}^{N} \left| \boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k \right|^2 \tag{9.101}$$

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 (9.102) and (9.96) we have that

$$P_{XX}(f_i) = 0, \qquad i = 1,\ \ldots,\ P \tag{9.102}$$

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

$$P_{XX}^{MUSIC}(f) = \frac{1}{\displaystyle\sum_{k=P+1}^{N} \left| \boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{v}_k \right|^2} = \frac{1}{\boldsymbol{s}^{\mathrm{H}}(f)\boldsymbol{V}(f)\boldsymbol{V}^{\mathrm{H}}(f)\boldsymbol{s}(f)} \tag{9.103}$$

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.

9.6.3 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:

$$y(m)=\sum_{k=1}^{P} A_k e^{-j(2\pi F_k m+\phi_k)}+n(m) \tag{9.104}$$

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

$$\boldsymbol{x}(m)=\boldsymbol{S}\,\boldsymbol{a} \tag{9.105}$$

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 (9.91) and (9.92). 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

$$\boldsymbol{x}(m+1)=\boldsymbol{S}\boldsymbol{\Phi}\boldsymbol{a} \tag{9.106}$$

where $\boldsymbol{\Phi}$ is a $P\times P$ phase matrix defined as

$$\boldsymbol{\Phi}=\mathrm{diag}[e^{j2\pi F_1}, e^{j2\pi F_2}, \cdots, e^{j2\pi F_P}] \tag{9.107}$$

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

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m)}=\boldsymbol{S}\boldsymbol{P}\boldsymbol{S}^{\mathrm{H}}+\sigma_n^2\mathbf{I} \tag{9.108}$$

where the matrix $\boldsymbol{P}$ is diagonal, and its diagonal elements are the powers of the complex sinusoids $\boldsymbol{P}=\text{diag}[A_1^2, \ldots, A_P^2] = \boldsymbol{a}\boldsymbol{a}^{\text{H}}$. The cross-covariance matrix of the vectors $\boldsymbol{y}(m)$ and $\boldsymbol{y}(m+1)$ is

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} = \boldsymbol{S}\boldsymbol{P}\boldsymbol{\Phi}^{\text{H}}\boldsymbol{S}^{\text{H}} + \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} \tag{9.109}$$

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

$$\boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} = \begin{pmatrix} r_{yy}(1) & r_{yy}(2) & r_{yy}(3) & \cdots & \mathrm{r}_{\mathrm{yy}}(N) \\ r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \cdots & r_{yy}(N-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \cdots & r_{yy}(N-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(N-2) & r_{yy}(N-3) & r_{yy}(N-4) & \cdots & r_{yy}(1) \end{pmatrix} \tag{9.110}$$

and

$$\boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \sigma_n^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_n^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 & 0 \end{pmatrix} \tag{9.111}$$

The correlation matrix of the signal vector $\boldsymbol{x}(m)$ can be estimated as

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m)} = \boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m)} - \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m)} = \boldsymbol{S}\boldsymbol{P}\boldsymbol{S}^{\text{H}} \tag{9.112}$$

and the cross-correlation matrix of the signal vector $\boldsymbol{x}(m)$ with its time-shifted version $\boldsymbol{x}(m+1)$ is obtained as

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m+1)} = \boldsymbol{R}_{\boldsymbol{y}(m)\boldsymbol{y}(m+1)} - \boldsymbol{R}_{\boldsymbol{n}(m)\boldsymbol{n}(m+1)} = \boldsymbol{S}\boldsymbol{P}\boldsymbol{\Phi}^{\text{H}}\boldsymbol{S}^{\text{H}} \tag{9.113}$$

Subtraction of a fraction $\lambda_i = e^{-j2\pi F_i}$ of Equation (9.113) from Equation (9.112) yields

$$\boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m)} - \lambda_i \boldsymbol{R}_{\boldsymbol{x}(m)\boldsymbol{x}(m+1)} = \boldsymbol{S}\boldsymbol{P}(\mathbf{I} - \lambda_i \boldsymbol{\Phi}^H)\boldsymbol{S}^H \tag{9.114}$$

From Equations (9.107) and (9.114), the frequencies of the sinusoids can be estimated as the roots of Equation (9.114).

9.7 Summary

Power spectrum estimation is perhaps the most widely used method of signal analysis. The main objective of any transformation is to express a signal in a form that lends itself to more convenient analysis and manipulation. The power spectrum is related to the correlation function through the Fourier transform. The power spectrum reveals the repetitive and correlated patterns of a signal, which are important in detection, estimation, data forecasting and decision-making systems. We began this chapter with Section 9.1 on basic definitions of the Fourier series/transform, energy spectrum and power spectrum. In Section 9.2, we considered 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 9.3, we considered the maximum-entropy and the model-based spectral estimation methods. These methods can extrapolate the correlation values beyond the range for which data is available, and hence can offer higher resolution and less side-lobes. In Section 9.4, 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**, pp. 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**, pp. 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, pp. 81–97.

CAPON J. (1969) High Resolution Frequency-Wavenumber Spectrum Analysis. Proc. IEEE. **57**, pp. 1408–1419.

CHILDERS D.G., Editor (1978) Modern Spectrum Analysis. IEEE Press.

COHEN L. (1989) Time-Frequency Distributions - A review. Proc. IEEE, **77**, pp. 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**, pp. 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**, pp. 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**, pp. 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**, pp. 329-349.

PISARENKO V.F. (1973) The Retrieval of Harmonics from a Covariance Function. Geophy. J. R. Astron. Soc., **33**, pp. 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., Editor (1986) Modern Spectrum Analysis. IEEE Press.

STRAND O.N. (1977) Multichannel Complex Maximum Entropy (AutoRegressive) Spectral Analysis. IEEE Trans. on Automatic Control, **22(4)**, pp. 634–640.

VAN DEN BOS A. (1971) Alternative Interpretation of Maximum Entropy Spectral Analysis. IEEE Trans. Infor. Tech., **IT-17**, pp. 92–99.

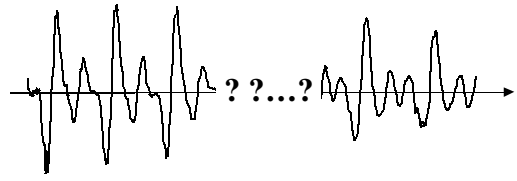
WELCH P.D. (1967) The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short Modified Periodograms. IEEE Trans. Audio and Electroacoustics, **AU-15**, pp. 70–79.

WILKINSON J.H. (1965) The Algebraic Eigenvalue Problem. Oxford University Press.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

10



INTERPOLATION

Interpolation is the estimation of the unknown, or the lost, samples of a signal using a weighted average of a number of known samples at the neighbourhood points. Interpolators are used in various forms in most 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, impulsive noise, dropouts, etc. This chapter begins with a study of the basic concept of ideal interpolation of a band-limited signal, a simple model for the effects of a number of missing samples, and the factors that affect the interpolation process. The classical approach to interpolation is to construct a polynomial that passes through the known samples. In Section 10.2, a general form of polynomial interpolation and its special forms, Lagrange, Newton, Hermite and cubic spline interpolators, are considered. Optimal interpolators utilise predictive and statistical models of the signal process. In Section 10.3, a number of model-based interpolation methods are considered. These methods include maximum a posteriori interpolation, and least square error interpolation based on an autoregressive model. Finally, we consider time–frequency interpolation, and interpolation through searching an adaptive signal codebook for the best-matching signal.

10.1 Introduction

The objective of interpolation is to obtain a high-fidelity reconstruction of the unknown or the missing samples of a signal. The emphasis in this chapter is on the interpolation of a *sequence* of lost samples. However, first in this section, the 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.

10.1.1 Interpolation of a Sampled Signal

A common application of interpolation is the reconstruction of a continuous-time signal $x(t)$ from a discrete-time signal $x(m)$. The condition for the recovery of a continuous-time signal from its samples is given by the Nyquist sampling theorem. The Nyquist theorem states that a band-limited signal, with a highest frequency content of F_c (Hz), can be reconstructed from its samples *if* the sampling speed is greater than $2F_c$ samples per second. Consider a band-limited continuous-time signal $x(t)$, sampled at a rate of F_s samples per second. The discrete-time signal $x(m)$ may be expressed as the following product:

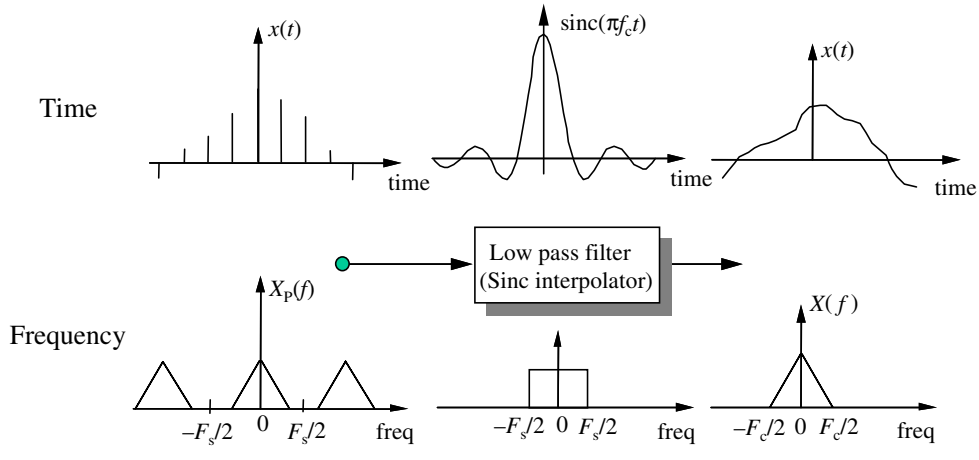


Figure 10.1 Reconstruction of a continuous-time signal from its samples. In frequency domain interpolation is equivalent to low-pass filtering.

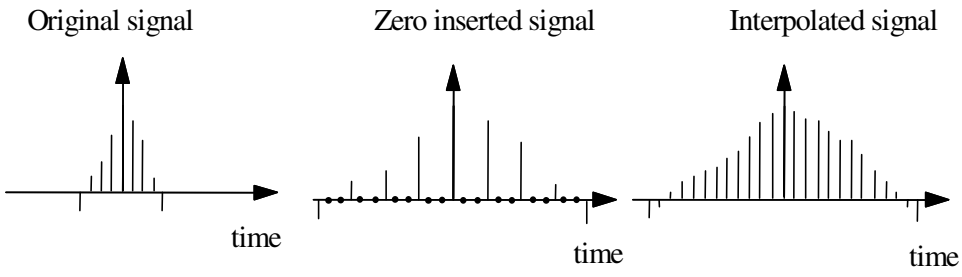


Figure 10.2 Illustration of up-sampling by a factor of 3 using a two-stage process of zero-insertion and digital low-pass filtering.

$$x(m)=x(t)\,p(t)=\sum_{m=-\infty}^{\infty}x(t)\delta(t-mT_s) \tag{10.1}$$

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 (10.1), it can be shown that the spectrum of the sampled signal is given by

$$X_s(f)=X(f)*P(f)=\sum_{k=-\infty}^{\infty}X(f+kf_s) \tag{10.2}$$

where $X(f)$ and $P(f)$ are the spectra of the signal $x(t)$ and the sampling function $p(t)$ respectively, and * denotes the convolution operation. Equation (10.2), illustrated in Figure 10.1, states that the spectrum of a sampled signal is composed of the original base-band spectrum $X(f)$ and the repetitions or images of $X(f)$ spaced uniformly at frequency intervals of $F_s=1/T_s$. When the sampling frequency is above the Nyquist rate, the base-band spectrum $X(f)$ is not overlapped by its images $X(f\pm kF_s)$, and the original signal can be recovered by a low-pass filter as shown in Figure 10.1. Hence the ideal interpolator of a band-limited discrete-time signal is an ideal low-pass filter with a sinc impulse response. The recovery of a continuous-time signal through sinc interpolation can be expressed as

$$x(t)=\sum_{m=-\infty}^{\infty}x(m)T_s f_c\,\mathrm{sinc}\left[\pi f_c(t-mT_s)\right] \tag{10.3}$$

In practice, the sampling rate F_s should be sufficiently greater than $2F_c$, say $2.5F_c$, in order to accommodate the transition bandwidth of the interpolating low-pass filter.

10.1.2 Digital Interpolation by a Factor of *I*

Applications of digital interpolators include sampling rate conversion in multirate communication systems and up-sampling for improved graphical representation. To change a sampling rate by a factor of $V=I/D$ (where I and D are integers), the signal is first interpolated by a factor of I, and then the interpolated signal is decimated by a factor of D.

Consider a band-limited discrete-time signal $x(m)$ with a base-band spectrum $X(f)$ as shown in Figure 10.2. The sampling rate can be increased by a factor of I through interpolation of I–1 samples between every two samples of $x(m)$. In the following it is shown that digital interpolation by a factor of I can be achieved through a two-stage process of (a) insertion of I–1 zeros in between every two samples and (b) low-pass filtering of the 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

$$x_z(m) = \begin{cases} x\left(\frac{m}{I}\right), & m = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{10.4}$$

The spectrum of the zero-inserted signal is related to the spectrum of the original discrete-time signal by

$$\begin{aligned} X_z(f) &= \sum_{m=-\infty}^{\infty} x_z(m) e^{-j2\pi fm} \\ &= \sum_{m=-\infty}^{\infty} x(m) e^{-j2\pi fmI} \\ &= X(I.f) \end{aligned} \tag{10.5}$$

Equation (10.5) states that the spectrum of the zero-inserted signal $X_z(f)$ is a frequency-scaled version of the spectrum of the original signal $X(f)$. Figure 10.2 shows that the base-band spectrum of the zero-inserted signal is composed of I repetitions of the based band spectrum of the original signal. The interpolation of the zero-inserted signal is therefore equivalent to filtering out the repetitions of $X(f)$ in the base band of $X_z(f)$, as illustrated in Figure 10.2. Note that to maintain the real-time duration of the signal the

sampling rate of the interpolated signal $x_z(m)$ needs to be increased by a factor of I.

10.1.3 Interpolation of a Sequence of Lost Samples

In this section, we introduce the problem of interpolation of a sequence of M missing samples of a signal given a number of samples on both side of the gap, as illustrated in Figure 10.3. Perfect interpolation 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.

A simple distortion model for a signal $y(m)$ with M missing samples, illustrated in Figure 10.3, is given by

$$\begin{aligned} y(m) &= x(m)d(m) \\ &= x(m)\,[1 - r(m)] \end{aligned} \tag{10.6}$$

where the distortion operator $d(m)$ is defined as

$$d(m) = 1 - r(m) \tag{10.7}$$

and $r(m)$ is a rectangular pulse of duration M samples starting at the sampling time k:

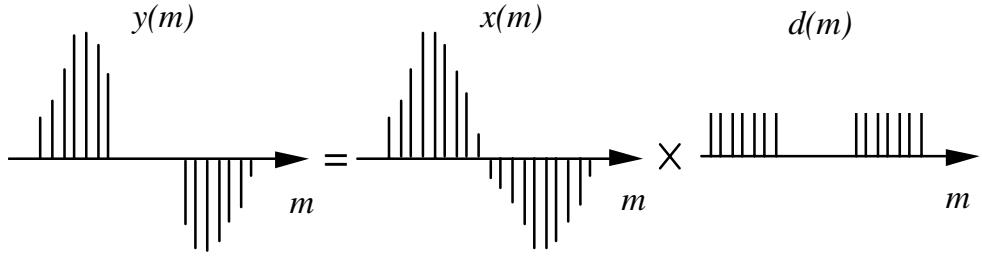


Figure 10.3 Illustration of a distortion model for a signal with a sequence of missing samples.

$$r(m)=\begin{cases}1, & k\le m\le k+M-1\\ 0, & \text{otherwise}\end{cases} \tag{10.8}$$

In the frequency domain, Equation (10.6) becomes

$$\begin{aligned}Y(f)&=X(f)*D(f)\\&=X(f)*[\delta(f)-R(f)]\\&=X(f)-X(f)*R(f)\end{aligned} \tag{10.9}$$

where $D(f)$ is the spectrum of the distortion $d(m)$, $\delta(f)$ is the Kronecker delta function, and $R(f)$, the frequency spectrum of the rectangular pulse $r(m)$, is given by

$$R(f)=e^{-j2\pi f[k+(M-1)/2]}\frac{\sin(\pi fM)}{\sin(\pi f)} \tag{10.10}$$

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 10.3, the interpolation error can be minimised through optimal utilisation of the signal models and the information contained in the neighbouring samples.

Example 10.1 *Interpolation of missing samples of a sinusoidal signal.* Consider a cosine waveform of amplitude A and frequency F_0 with M missing samples, modelled as

$$\begin{aligned}y(m)&=x(m)\,d(m)\\&=A(\cos 2\pi f_0 m)[1-r(m)b]\end{aligned} \tag{10.11}$$

where $r(m)$ is the rectangular pulse defined in Equation (10.7). In the frequency domain, the distorted signal can be expressed as

$$\begin{aligned}Y(f)&=\frac{A}{2}[\delta(f-f_o)+\delta(f+f_o)]*[\delta(f)-R(f)]\\&=\frac{A}{2}[\delta(f-f_o)+\delta(f+f_o)-R(f-f_o)-R(f+f_o)]\end{aligned} \tag{10.12}$$

where $R(f)$ is the spectrum of the pulse $r(m)$ as in Equation (10.9).

From Equation (10.12), it is evident that, for a cosine signal of frequency F_0, the distortion in the frequency domain due to the missing samples is manifested in the appearance of sinc functions centred at $\pm F_0$. The distortion can be removed by filtering the signal with a very narrow band-pass filter. Note that for a cosine signal, perfect restoration is possible only because the signal has infinitely narrow bandwidth, or equivalently because the signal is completely predictable. In fact, for this example, the distortion can also be removed using a linear prediction model, which, for a cosine signal, can be regarded as a data-adaptive narrow band-pass filter.

10.1.4 The Factors That Affect Interpolation Accuracy

The interpolation accuracy is affected by a number of factors, the most important of which are as follows:

(a) 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.
(b) The sampling rate: as the sampling rate increases, adjacent samples become more correlated, the redundant information increases, and interpolation improves.
(c) 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.
(d) The length of the missing samples: in general, interpolation quality decreases with increasing length of the missing samples.
(e) 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 10.3, we study several statistical digital signal processing methods for interpolation of a sequence of missing samples. These include model-based methods, which are well suited for interpolation of small to medium sized gaps of missing samples. We also consider frequency–time interpolation methods, and interpolation through waveform substitution, which have the ability to replace relatively large gaps of missing samples.

10.2 Polynomial Interpolation

The classical approach to interpolation is to construct a polynomial interpolator that passes through the known samples. Polynomial interpolators may be formulated in various forms, such as power series, Lagrange interpolation and Newton interpolation. These various forms are mathematically equivalent and can be transformed from one into another. Suppose the data consists of N+1 samples $\{x(t_0), x(t_1), ..., x(t_N)\}$, where $x(t_n)$ denotes the amplitude of the signal $x(t)$ at time t_n. The polynomial of order N that passes through the N+1 known samples is unique (Figure 10.4) and may be written in power series form as

$$\hat{x}(t) = p_N(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_N t^N \tag{10.13}$$

where $P_N(t)$ is a polynomial of order N, and the a_k are the polynomial coefficients. From Equation (10.13), and a set of N+1 known samples, a

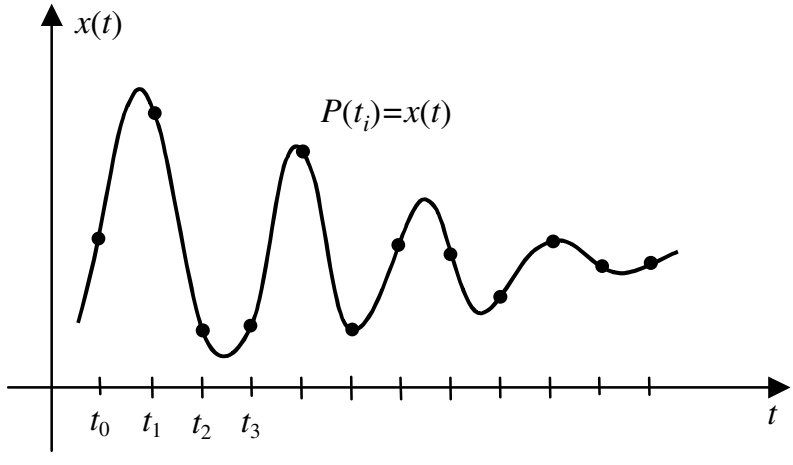


Figure 10.4 Illustration of an Interpolation curve through a number of samples.

system of N+1 linear equations with N+1 unknown coefficients can be formulated as

$$\begin{aligned} x(t_0) &= a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + \cdots + a_N t_0^N \\ x(t_1) &= a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + \cdots + a_N t_1^N \\ &\vdots \qquad\quad \vdots \qquad\quad \ddots \\ x(t_N) &= a_0 + a_1 t_N + a_2 t_N^2 + a_3 t_N^3 + \cdots + a_N t_N^N \end{aligned} \tag{10.14}$$

From Equation (10.14). the polynomial coefficients are given by

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} 1 & t_0 & t_0^2 & t_0^3 & \cdots & t_0^N \\ 1 & t_1 & t_1^2 & t_1^3 & \cdots & t_1^N \\ 1 & t_2 & t_2^2 & t_2^3 & \cdots & t_2^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & t_N^2 & t_N^3 & \cdots & t_N^N \end{pmatrix}^{-1} \begin{pmatrix} x(t_0) \\ x(t_1) \\ x(t_2) \\ \vdots \\ x(t_N) \end{pmatrix} \tag{10.15}$$

The matrix in Equation (10.15) is called a Vandermonde matrix. For a large number of samples, N, the Vandermonde matrix becomes large and ill-conditioned. An ill-conditioned matrix is sensitive to small computational errors, such as quantisation errors, and can easily produce inaccurate results. There are alternative methods of implementation of the polynomial interpolator that are simpler to program and/or better structured, such as Lagrange and Newton methods. However, it must be noted that these variants of the polynomial interpolation also become ill-conditioned for a large number of samples, N.

10.2.1 Lagrange Polynomial Interpolation

To introduce the Lagrange interpolation, consider a line interpolator passing through two points $x(t_0)$ and $x(t_1)$:

$$\hat{x}(t) = p_1(t) = x(t_0) + \underbrace{\frac{x(t_1) - x(t_0)}{t_1 - t_0}}_{\text{line slope}} (t - t_0) \tag{10.16}$$

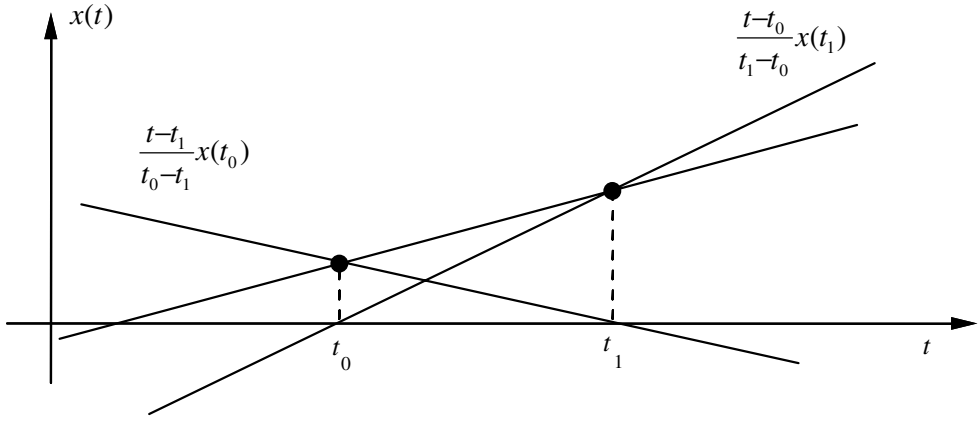


Figure 10.5 The Lagrange line interpolator passing through $x(t_0)$ and $x(t_1)$, described in terms of the combination of two lines: one passing through $(x(t_0), t_1)$ and the other through $(x(t_1), t_0)$.

The line Equation (10.16) may be rearranged and expressed as

$$p_1(t) = \frac{t-t_1}{t_0-t_1}\,x(t_0) + \frac{t-t_0}{t_1-t_0}\,x(t_1) \tag{10.17}$$

Equation (10.17) is in the form of a Lagrange polynomial. Note that the Lagrange form of a line interpolator is composed of the weighted combination of two lines, as illustrated in Figure 10.5.
In general, the Lagrange polynomial, of order N, passing through N+1 samples $\{x(t_0), x(t_1), \ldots x(t_N)\}$ is given by the polynomial equation

$$P_N(t) = L_0(t)x(t_0) + L_1(t)x(t_1) + \cdots + L_N(t)x(t_N) \tag{10.18}$$

where each Lagrange coefficient $L_N(t)$ is itself a polynomial of degree N given by

$$L_i(t) = \frac{(t-t_0)\cdots(t-t_{i-1})(t-t_{i+1})\cdots(t-t_N)}{(t_i-t_0)\cdots(t_i-t_{i-1})(t_i-t_{i+1})\cdots(t_i-t_N)} = \prod_{\substack{n=0 \\ n\neq i}}^{N} \frac{t-t_n}{t_i-t_n} \tag{10.19}$$

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:

(a) The computational complexity is large.
(b) The coefficients of a polynomial of order N cannot be used in the calculations of the coefficients of a higher order polynomial.
(c) The evaluation of the interpolation error is difficult.

The Newton polynomial, introduced in the next section, overcomes some of these difficulties.

10.2.2 Newton Polynomial Interpolation

Newton polynomials have a recursive structure, such that a polynomial of order N can be constructed by extension of a polynomial of order N–1 as follows:

$$
\begin{aligned}
p_0(t) &= a_0 && \text{(d.c. value)} \\
p_1(t) &= a_0 + a_1(t-t_0) \\
&= p_0(t) + a_1(t-t_0) && \text{(ramp)} \\
p_2(t) &= \underbrace{a_0 + a_1(t-t_0)} + a_2(t-t_0)(t-t_1) \\
&= \quad p_1(t) \quad + a_2(t-t_0)(t-t_1) && \text{(quadratic)} \\
p_3(t) &= \underbrace{a_0 + a_1(t-t_0) + a_2(t-t_0)(t-t_1)} + a_3(t-t_0)(t-t_1)(t-t_2) \\
&= \qquad\qquad p_2(t) \qquad\qquad + a_3(t-t_0)(t-t_1)(t-t_2) && \text{(cubic)}
\end{aligned}
\tag{10.20}
$$

and in general the recursive, *order update*, form of a Newton polynomial can be formulated as

$$p_N(t) = p_{N-1}(t) + a_N(t-t_0)(t-t_1)\cdots(t-t_{N-1}) \tag{10.21}$$

For a sequence of N+1 samples $\{x(t_0), x(t_1), \ldots x(t_N)\}$, 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 (10.21) at $t=t_0$ to $x(t_0)$:

$$p_N(t_0)= p_0(t_0)=x(t_0)=a_0 \tag{10.22}$$

To solve for the coefficient a_1, the first-order polynomial $p_1(t)$ is evaluated at $t=t_1$:

$$p_1(t_1)=x(t_1)=a_0+a_1(t_1-t_0)=x(t_0)+a_1(t_1-t_0) \tag{10.23}$$

from which

$$a_1=\frac{x(t_1)-x(t_0)}{t_1-t_0} \tag{10.24}$$

Note that the coefficient a_1 is the slope of the line passing through the points $[x(t_0), x(t_1)]$. To solve for the coefficient a_2 the second-order polynomial $p_2(t)$ is evaluated at $t=t_2$:

$$p_2(t_2)=x(t_2)=a_0+a_1(t_2-t_0)+a_2(t_2-t_0)(t_2-t_1) \tag{10.25}$$

Substituting a_0 and a_1 from Equations (10.22) and (10.24) in Equation (10.25) we obtain

$$a_2=\left[\frac{x(t_2)-x(t_1)}{t_2-t_1}-\frac{x(t_1)-x(t_0)}{t_1-t_0}\right]\Big/(t_2-t_0) \tag{10.26}$$

Each term in the square brackets of Equation (10.26) is a slope term, and the coefficient a_2 is the slope of the slope. To formulate a solution for the higher-order coefficients, we need to introduce the concept of *divided differences*. Each of the two ratios in the square brackets of Equation (10.26) is a so-called "divided difference". The divided difference between two points t_i and t_{i-1} is defined as

$$d_1(t_{i-1},t_i)=\frac{x(t_i)-x(t_{i-1})}{t_i-t_{i-1}} \tag{10.27}$$

The divided difference between two points may be interpreted as the average difference or the slope of the line passing through the two points. The second-order divided difference (i.e. the divided difference of the divided difference) over three points t_{i-2} , t_{i-1} and t_i is given by

$$d_2(t_{i-2},t_i)=\frac{d_1(t_{i-1},t_i)-d_1(t_{i-2},t_{i-1})}{t_i-t_{i-2}} \tag{10.28}$$

and the third-order divided difference is

$$d_3(t_{i-3},t_i)=\frac{d_2(t_{i-2},t_i)-d_2(t_{i-3},t_{i-1})}{t_i-t_{i-3}} \tag{10.29}$$

and so on. In general the j^{th} order divided difference can be formulated in terms of the divided differences of order j–1, in an order-update equation given as

$$d_j(t_{i-j},t_i)=\frac{d_{j-1}(t_{i-j+1},t_i)-d_{j-1}(t_{i-j},t_{i-1})}{t_i-t_{i-j}} \tag{10.30}$$

Note that $a_1=d_1(t_0,t_1)$, $a_2=d_2(t_0,t_2)$ and $a_3=d_3(t_0,t_3)$, and in general the Newton polynomial coefficients are obtained from the divided differences using the relation

$$a_i=d_i(t_0,t_i) \tag{10.31}$$

A main advantage of the Newton polynomial is its computational efficiency, in that a polynomial of order N–1 can be easily extended to a higher-order polynomial of order N. This is a useful property in the selection of the best polynomial order for a given set of data.

10.2.3 Hermite Polynomial Interpolation

Hermite polynomials are formulated to fit not only to the signal samples, but also to the derivatives of the signal as well. Suppose the data consists of N+1 samples and assume that all the derivatives up to the M^{th} order derivative are available. Let the data set, i.e. the signal samples and the derivatives, be denoted as $[x(t_i),x'(t_i),x''(t_i),\ldots,x^{(M)}(t_i),i=0,\ldots,N]$. There

are altogether $K=(N+1)(M+1)$ data points and a polynomial of order $K-1$ can be fitted to the data as

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \cdots + a_{K-1} t^{K-1} \tag{10.32}$$

To obtain the polynomial coefficients, we substitute the given samples in the polynomial and its M derivatives as

$$\begin{aligned} p(t_i) &= x(t_i) \\ p'(t_i) &= x'(t_i) \\ p''(t_i) &= x''(t_i) \\ \vdots &= \vdots \\ p^{(M)}(t_i) &= x^{(M)}(t_i), \qquad i=0,1,\ldots,N \end{aligned} \tag{10.33}$$

In all, there are $K=(M+1)(N+1)$ equations in (10.33), and these can be used to calculate the coefficients of the polynomial Equation (10.32). In theory, the constraint that the polynomial must also fit the derivatives should result in a better interpolating polynomial that passes through the sampled points and is also consistent with the known underlying dynamics (i.e. the derivatives) of the curve. However, even for moderate values of N and M, the size of Equation (10.33) becomes too large for most practical purposes.

10.2.4 Cubic Spline Interpolation

A polynomial interpolator of order N is constrained to pass through $N+1$ known samples, and can have $N-1$ maxima and minima. In general, the interpolation error increases rapidly with the increasing polynomial order, as the interpolating curve has to wiggle through the $N+1$ samples. When a large number of samples are to be fitted with a smooth curve, it may be better to divide the signal into a number of smaller intervals, and to fit a low order interpolating polynomial to each small interval. Care must be taken to ensure that the polynomial curves are continuous at the endpoints of each interval. In cubic spline interpolation, a cubic polynomial is fitted to each interval between two samples. A cubic polynomial has the form

$$p(t)=a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{10.34}$$

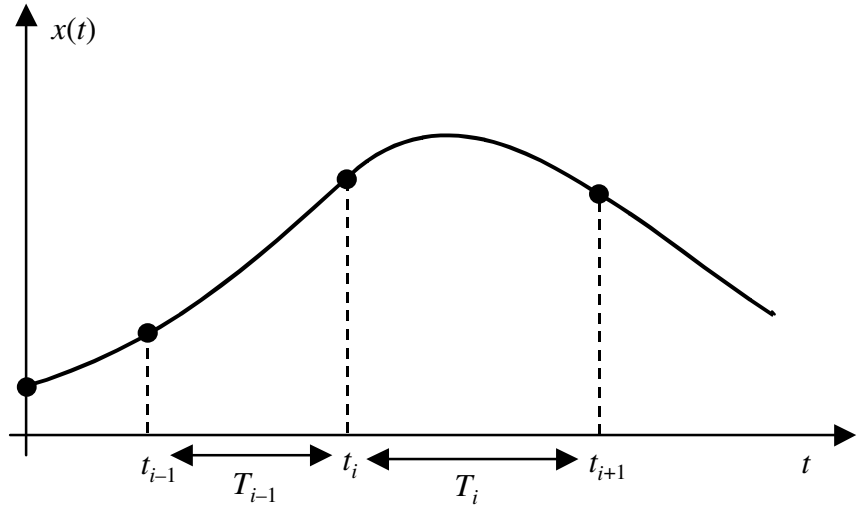


Figure 10.6 Illustration of cubic spline interpolation.

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 10.6. Using a local coordinate $\tau = t - t_i$, the cubic polynomial becomes

$$p(\tau) = a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 \tag{10.35}$$

At $\tau = 0$, we obtain the first coefficient a_0 as

$$a_0 = p(\tau = 0) = x(t_i) \tag{10.36}$$

The second derivative of $p(\tau)$ is given by

$$p''(\tau) = 2a_2 + 6a_3\tau \tag{10.37}$$

Evaluation of the second derivative at $\tau = 0$ (i.e. $t = t_i$) gives the coefficient a_2

$$a_2 = \frac{p_i''(\tau = 0)}{2} = \frac{p_i''}{2} \tag{10.38}$$

Similarly, evaluating the second derivative at the point t_{i+1} (i.e. $\tau = T_i$) yields the fourth coefficient

$$a_3 = \frac{p_{i+1}'' - p_i''}{6T_i} \tag{10.39}$$

Now to obtain the coefficient a_1, we evaluate $p(\tau)$ at $\tau = T_i$:

$$p(\tau = T_i) = a_0 + a_1 T_i + a_2 T_i^{\,2} + a_3 T_i^{\,3} = x(t_{i+1}) \tag{10.40}$$

and substitute a_0, a_2 and a_3 from Equations (10.36), (10.38) and (10.39) in (10.40) to obtain

$$a_1 = \frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p_{i+1}'' + 2p_i''}{6} T_i \tag{10.41}$$

The cubic polynomial can now be written as

$$p(\tau) = x(t_i) + \left[\frac{x(t_{i+1}) - x(t_i)}{T_i} - \frac{p_{i+1}'' + 2p_i''}{6} T_i \right] \tau + \frac{p_i''}{2} \tau^2 + \frac{p_{i+1}'' - p_i''}{6T_i} \tau^3 \tag{10.42}$$

To determine the coefficients of the polynomial in Equation (10.42), we need the second derivatives and p_{i+1}''. These are obtained from the constraint that the first derivatives of the curves at the endpoints of each interval must be continuous. From Equation (10.42), the first derivatives of $p(\tau)$ evaluated at the endpoints t_i and t_{i+1} are

$$p_i' = p'(\tau = 0) = -\frac{T_i}{6}[p_{i+1}'' + 2p_i''] + \frac{1}{T_i}[x(t_{i+1}) - x(t_i)] \tag{10.43}$$

$$p_{i+1}' = p'(\tau = T_i) = \frac{T_i}{6}[2p_{i+1}'' + p_i''] + \frac{1}{T_i}[x(t_{i+1}) - x(t_i)] \tag{10.44}$$

Similarly, for the preceding interval, $t_{i-1}<t<t_i$, the first derivative of the cubic spline curve evaluated at $\tau=t_i$ is given by

$$p'_i = p'(\tau = t_i) = \frac{T_{i-1}}{6}[2p''_i + p''_{i-1}] + \frac{1}{T_{i-1}}[x(t_i) - x(t_{i-1})] \qquad (10.45)$$

For continuity of the first derivative at t_i, p'_i at the end of the interval (t_{i-1} ,t_i) must be equal to the p'_i at the start of the interval (t_i ,t_{i+1}). Equating the right-hand sides of Equations (10.43) and (10.45) and repeating this exercise yields

$$T_{i-1}\,p''_{i-1} + 2(T_{i-1}+T_i)\,p''_i + T_i\,p''_{i+1} = 6\left[\frac{1}{T_{i-1}}x(t_{i-1}) - \left(\frac{1}{T_{i-1}} + \frac{1}{T_i}\right)x(t_i) + \frac{1}{T_i}x(t_{i+1})\right]$$
$$i = 1, 2, \ldots, N-1 \qquad (10.46)$$

In Equation (10.46), there are N–1 equations in N+1 unknowns p''_i. For a unique solution we need to specify the second derivatives at the points t_0 and t_N. This can be done in two ways: (a) setting the second derivatives at the endpoints t_0 and t_N (i.e. p''_0 and p''_N), to zero, or (b) extrapolating the derivatives from the inside data.

10.3 Model-Based Interpolation

The statistical signal processing approach to interpolation of a sequence of lost samples is based on the utilisation of a predictive and/or a probabilistic model of the signal. In this section, we study the maximum a posteriori interpolation, an autoregressive model-based interpolation, a frequency–time interpolation method, and interpolation through searching a signal record for the best replacement.

Figures 10.7 and 10.8 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, $\boldsymbol{x}_{\mathrm{Uk}}=\{x(k), x(k+1), ..., x(k+M-1)\}$ 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 process. An N-sample signal vector

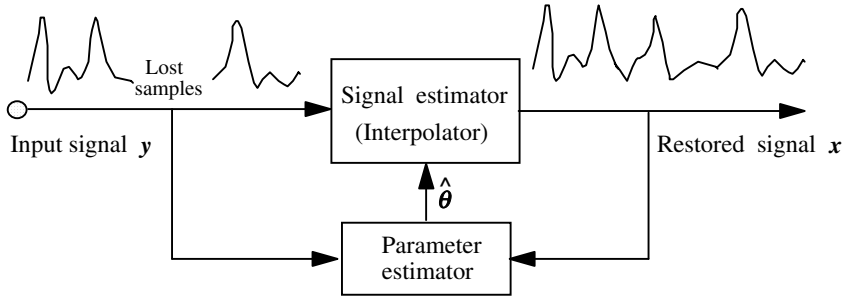


Figure 10.7 Illustration of a model-based iterative signal interpolation system.

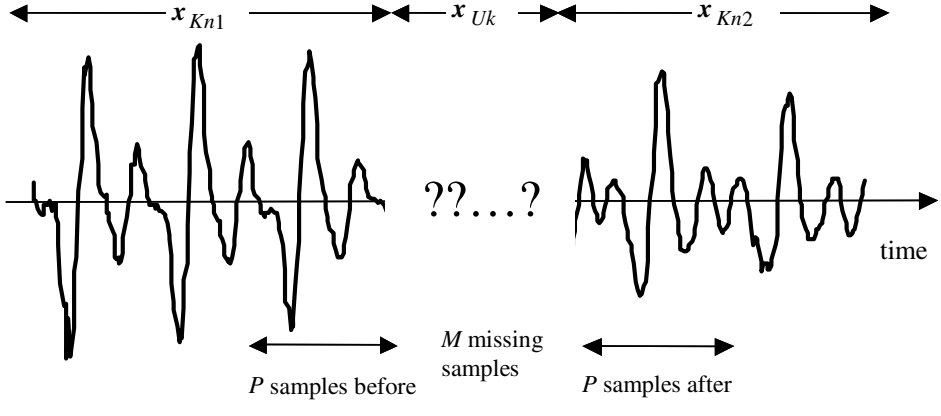


Figure 10.8 A signal with *M* missing samples and *N–M* known samples. On each side of the missing segment, *P* samples are used to interpolate the segment.

$\boldsymbol{x}$, composed of M unknown samples and $N–M$ known samples, can be written as

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_{Kn_1} \\ \boldsymbol{x}_U \\ \boldsymbol{x}_{Kn_2} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_{Kn_1} \\ \boldsymbol{0} \\ \boldsymbol{x}_{Kn_2} \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{x}_{Uk} \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{K}\,\boldsymbol{x}_{Kn} + \boldsymbol{U}\,\boldsymbol{x}_{Uk} \tag{10.47}$$

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 10.8. The matrices $\boldsymbol{K}$ and $\boldsymbol{U}$ in Equation (10.47) are rearrangement matrices that assemble the vector $\boldsymbol{x}$ from $\boldsymbol{x}_{\mathrm{Kn}}$ and $\boldsymbol{x}_{\mathrm{Uk}}$.

10.3.1 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

$$\begin{aligned} f_X(\boldsymbol{x}_{\mathrm{Uk}}|\boldsymbol{x}_{\mathrm{Kn}}) &= \frac{f_X(\boldsymbol{x}_{\mathrm{Kn}},\boldsymbol{x}_{\mathrm{Uk}})}{f_X(\boldsymbol{x}_{\mathrm{Kn}})} \\ &= \frac{f_X(\boldsymbol{x}=\boldsymbol{K}\boldsymbol{x}_{\mathrm{Kn}}+\boldsymbol{U}\boldsymbol{x}_{\mathrm{Uk}})}{f_X(\boldsymbol{x}_{\mathrm{Kn}})} \end{aligned} \tag{10.48}$$

In Equation (10.48), 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

$$\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{MAP} = \arg\max_{\boldsymbol{x}_{\mathrm{Uk}}} f_X(\boldsymbol{K}\boldsymbol{x}_{\mathrm{Kn}}+\boldsymbol{U}\boldsymbol{x}_{\mathrm{Uk}}) \tag{10.49}$$

Example 10.2 *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 (10.47). The pdf of the signal $\boldsymbol{x}$ is given by

$$f_X(\boldsymbol{x}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{x}\right) \tag{10.50}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ is the covariance matrix of the Gaussian vector process $\boldsymbol{x}$. Substitution of Equation (10.50) in Equation (10.48) yields the conditional pdf of the unknown signal $\boldsymbol{x}_{\mathrm{Uk}}$ given a number of samples $\boldsymbol{x}_{\mathrm{Kn}}$:

$$\begin{aligned} f_X(\boldsymbol{x}_{Uk}|\boldsymbol{x}_{Kn}) = \frac{1}{f_X(\boldsymbol{x}_{\mathrm{Kn}})}\frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}_{\boldsymbol{xx}}|^{1/2}}\times \\ \exp\left(-\frac{1}{2}(\boldsymbol{K}\boldsymbol{x}_{\mathrm{Kn}}+\boldsymbol{U}\boldsymbol{x}_{\mathrm{Uk}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}(\boldsymbol{K}\boldsymbol{x}_{\mathrm{Kn}}+\boldsymbol{U}\boldsymbol{x}_{\mathrm{Uk}})\right) \end{aligned} \tag{10.51}$$

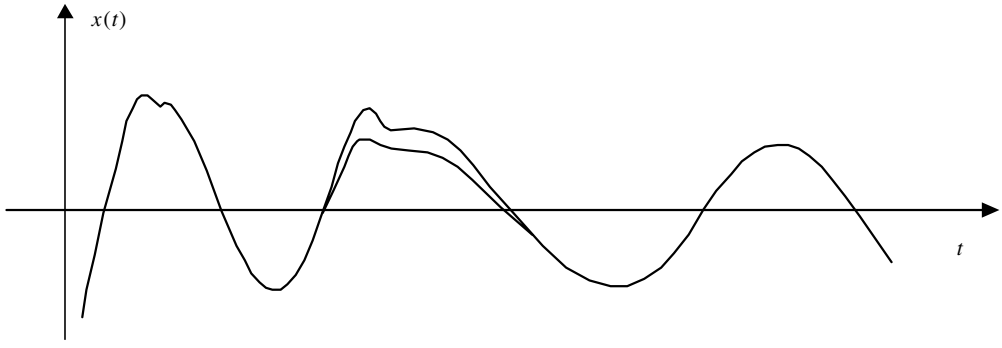


Figure 10.9 Illustration of MAP interpolation of a segment of 20 samples.

The MAP signal estimate, obtained by setting the derivative of the log-likelihood function $\ln f_X(\boldsymbol{x}|\boldsymbol{x}_{\text{Kn}})$ of Equation (10.51) with respect to $\boldsymbol{x}_{\text{Uk}}$ to zero, is given by

$$\boldsymbol{x}_{\text{Uk}} = -\left(\boldsymbol{U}^{\text{T}} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{U}\right)^{-1} \boldsymbol{U}^{\text{T}} \boldsymbol{\Sigma}_{xx}^{-1} \boldsymbol{K} \boldsymbol{x}_{\text{Kn}} \tag{10.52}$$

An example of MAP interpolation is shown in Figure 10.9.

10.3.2 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 7. 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 10.3.4, the AR interpolation method is extended to include pitch–period correlations.

10.3.3 Interpolation Based on a Short-Term Prediction Model

An autoregressive (AR), or linear predictive, signal $x(m)$ is described as

$$x(m) = \sum_{k=1}^{P} a_k\, x(m-k) + e(m) \tag{10.53}$$

where $x(m)$ is the AR signal, a_k are the model coefficients and $e(m)$ is a zero mean excitation signal. The excitation may be a random signal, a 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, $\boldsymbol{x}_{\mathrm{Uk}}=\{x(k), ..., x(k+M-1)\}$ are missing. The objective is to estimate the missing samples $\boldsymbol{x}_{\mathrm{Uk}}$, using the remaining $N-k$ samples and an AR model of the signal. Figure 10.8 illustrates the interpolation problem. For this signal record of N samples, the AR equation (10.53) can be expanded to form the following matrix equation:

$$\begin{pmatrix} e(P) \\ e(P+1) \\ \vdots \\ e(k-1) \\ \hline e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \\ \hline e(k+M+P) \\ e(k+M+P+1) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(P) \\ x(P+1) \\ \vdots \\ x(k-1) \\ \hline x_{\mathrm{Uk}}(k) \\ x_{\mathrm{Uk}}(k+1) \\ x_{\mathrm{Uk}}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \\ \hline x(k+M+P) \\ x(k+M+P+1) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(P-1) & x(P-2) & \dots & x(0) \\ x(P) & x(P-1) & \dots & x(1) \\ \vdots & \vdots & \ddots & \vdots \\ x(k-2) & x(k-3) & \dots & x(k-P-1) \\ \hline x(k-1) & x(k-2) & \dots & x(k-P) \\ x_{Uk}(k) & x(k-1) & \dots & x(k-P+1) \\ x_{Uk}(k+1) & x_{Uk}(k) & \dots & x(k-P+2) \\ \vdots & \vdots & \ddots & \vdots \\ x(k+M+P-3) & x(k+M+P-2) & \dots & x_{\mathrm{Uk}}(k+M-2) \\ x(k+M+P-2) & x(k+M+P-1) & \dots & x_{\mathrm{Uk}}(k+M-1) \\ \hline x(k+M+P-1) & x(k+M+P) & \dots & x(k+M) \\ x(k+M+P) & x(k+M+P+1) & \dots & x(k+M+1) \\ \dots & \dots & \ddots & \dots \\ x(N-2) & x(N-3) & \dots & x(N-P-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} \tag{10.54}$$

where the subscript Uk denotes the unknown samples. Equation (10.54) can be rewritten in compact vector notation as

$$\boldsymbol{e}(\boldsymbol{x}_{\mathrm{Uk}}, \boldsymbol{a}) = \boldsymbol{x} - \boldsymbol{X}\boldsymbol{a} \tag{10.55}$$

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 criteriobbn 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

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}(\boldsymbol{x}_{\mathrm{Uk}}, \boldsymbol{a}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{a} - 2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{x} \tag{10.56}$$

The squared error function in Equation (10.56) 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 6.7. In the first stage of the solution, Equation (10.54) is *linearised* by either assuming that the missing samples have zero values or discarding the set of equations in (10.54), between the two dashed lines, that involve the unknown signal samples. The linearised equations are used to solve for the AR model coefficient vector $\boldsymbol{a}$ by forming the equation

$$\hat{\boldsymbol{a}} = \left(\boldsymbol{X}_{\mathrm{Kn}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{Kn}}\right)^{-1}\left(\boldsymbol{X}_{\mathrm{Kn}}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{Kn}}\right) \tag{10.57}$$

where the vector is an estimate of the model coefficients, obtained from the available signal samples.

The second stage of the solution involves the estimation of the unknown signal samples $\boldsymbol{x}_{\mathrm{Uk}}$. For an AR model of order P, and an unknown signal segment of length M, there are $2M+P$ nonlinear equations in (10.54) that involve the unknown samples; these are

$$\begin{pmatrix} e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \end{pmatrix} = \begin{pmatrix} x_{\mathrm{Uk}}(k) \\ x_{\mathrm{Uk}}(k+1) \\ x_{\mathrm{Uk}}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \end{pmatrix} - \begin{pmatrix} x(k-1) & x(k-2) & \cdots & x(k-p) \\ x_{\mathrm{Uk}}(k) & x(k-1) & \cdots & x(k-p+1) \\ x_{\mathrm{Uk}}(k+1) & x_{\mathrm{Uk}}(k) & \cdots & x(k-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ x_{\mathrm{Uk}}(k+M+P-3) & x_{\mathrm{Uk}}(k+M+P-4) & \cdots & x_{\mathrm{Uk}}(k+M-2) \\ x_{\mathrm{Uk}}(k+M+P-2) & x_{\mathrm{Uk}}(k+M+P-3) & \cdots & x_{\mathrm{Uk}}(k+M-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{P-1} \\ a_P \end{pmatrix} \tag{10.58}$$

The estimate of the predictor coefficient vector , obtained from the first stage of the solution, is substituted in Equation (10.58) so that the only remaining unknowns in (10.58) are the missing signal samples. Equation (10.58) may be partitioned and rearranged in vector notation in the following form:

$$\begin{pmatrix} e(k) \\ e(k+1) \\ e(k+2) \\ e(k+3) \\ e(k+4) \\ \vdots \\ e(k+P-1) \\ e(k+P) \\ e(k+P+1) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -a_1 & 1 & 0 & 0 & \cdots & 0 \\ -a_2 & -a_1 & 1 & 0 & \cdots & 0 \\ -a_3 & -a_2 & -a_1 & 1 & \cdots & 0 \\ -a_4 & -a_3 & -a_2 & -a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_P & -a_{P-1} & -a_{P-2} & -a_{P-3} & \cdots & 0 \\ 0 & -a_P & -a_{P-1} & -a_{P-2} & \cdots & 0 \\ 0 & 0 & -a_P & -a_{P-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -a_{P-1} \\ 0 & 0 & 0 & 0 & \cdots & -a_P \end{pmatrix} \begin{pmatrix} x_{\mathrm{Uk}}(k) \\ x_{\mathrm{Uk}}(k+1) \\ x_{\mathrm{Uk}}(k+2) \\ x_{\mathrm{Uk}}(k+3) \\ \vdots \\ x_{\mathrm{Uk}}(k+M-1) \end{pmatrix} +$$

$$\begin{pmatrix} -a_P & -a_{P-1} & -a_{P-2} & \cdots & -a_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & -a_P & -a_{P-1} & \cdots & -a_2 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & -a_P & \cdots & -a_3 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -a_P & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_2 & -a_1 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_3 & -a_2 & -a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & -a_{P-1} & -a_{P-2} & -a_{P-3} & \cdots & -a_1 \end{pmatrix} \begin{pmatrix} x(k-P) \\ x(k-P+1) \\ x(k-P+2) \\ \vdots \\ x(k-1) \\ 0 \\ \vdots \\ x(k+M) \\ x(k+M+1) \\ x(k+M+2) \\ \vdots \\ x(k+M+P-1) \end{pmatrix} \tag{10.59}$$

In Equation (10.59), the unknown and known samples are rearranged and grouped into two separate vectors. In a compact vector–matrix notation, Equation (10.58) can be written in the form

$$\boldsymbol{e} = \boldsymbol{A}_1 \boldsymbol{x}_{\mathrm{Uk}} + \boldsymbol{A}_2 \boldsymbol{x}_{\mathrm{Kn}} \tag{10.60}$$

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 (10.58). The total squared error is given by

$$\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}=(\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}}+\boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}})^{\mathrm{T}}(\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Uk}}+\boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}}) \tag{10.61}$$

The least square AR (LSAR) interpolation is obtained by minimisation of the squared error function with respect to the unknown signal samples $\boldsymbol{x}_{\mathrm{Uk}}$:

$$\frac{\partial \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{\partial \boldsymbol{x}_{\mathrm{Uk}}}=2\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{A}_1\boldsymbol{x}_{\mathrm{Kn}}+2\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{A}_2\boldsymbol{x}_{\mathrm{Kn}}=0 \tag{10.62}$$

From Equation (10.62) we have

$$\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{LSAR}=-\left(\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{A}_1\right)^{-1}\left(\boldsymbol{A}_1^{\mathrm{T}}\boldsymbol{A}_2\right)\boldsymbol{x}_{\mathrm{Kn}} \tag{10.63}$$

The solution in Equation (10.62) gives the $\hat{\boldsymbol{x}}_{\mathrm{Uk}}^{LSAR}$ vector which is the least square error estimate of the unknown data vector.

10.3.4 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 (10.53),

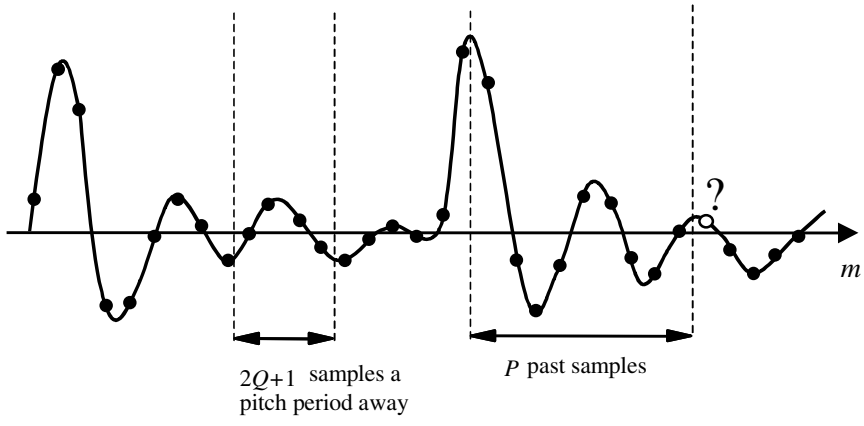


Figure 10.10 A quasiperiodic waveform. The sample marked " ? " is predicted using P immediate past samples and $2Q+1$ samples a pitch period away.

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 10.10. This modified AR model may be expressed by the following equation:

$$x(m) = \sum_{k=1}^{P} a_k x(m-k) + \sum_{k=-Q}^{Q} p_k x(m-T-k) + e(m) \qquad (10.64)$$

The AR model of Equation (10.64) is composed of a *short-term predictor* $\Sigma a_k\, x(m\text{-}k)$ that models the contribution of the P immediate past samples, and a *long-term predictor* $\Sigma p_k\, x(m–T–k)$ 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 (10.64) is specified by the parameter vector $\boldsymbol{c}=[a_1, ..., a_P, p_{-Q}, ..., p_Q]$ and the pitch period T. Note that in Figure 10.10

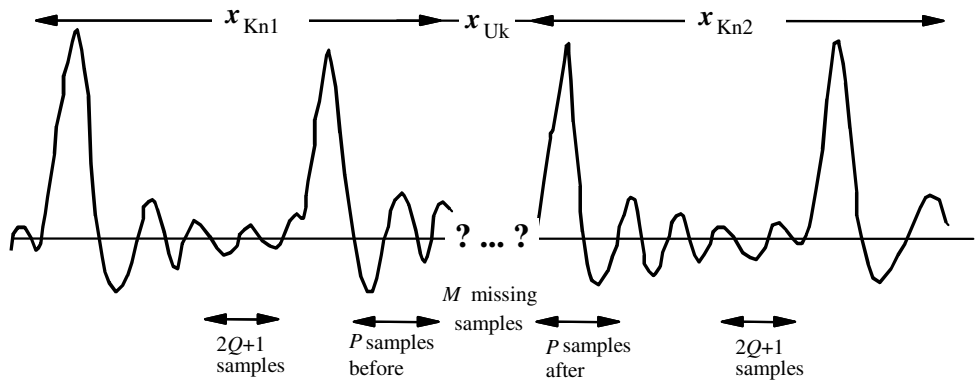


Figure 10.11 A signal with *M* missing samples. *P* immediate samples each side of the gap and 2*Q*+1 samples a pitch period away are used for interpolation.

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 pulse event. The sample is more predictable from the 2*Q*+1 samples a pitch period away, since they include the effects of a similar excitation pulse. The predictor coefficients are estimated (see Chapter 7) using the so-called normal equations:

$$\boldsymbol{c} = \boldsymbol{R}_{\boldsymbol{xx}}^{-1}\boldsymbol{r}_{\boldsymbol{xx}} \tag{10.65}$$

where $\boldsymbol{R}_{\boldsymbol{xx}}$ is the autocorrelation matrix of signal $\boldsymbol{x}$ and $\boldsymbol{r}_{\mathbf{xx}}$ is the correlation vector. In expanded form, Equation (10.65) can be written as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ p_{-Q+1} \\ \vdots \\ p_{+Q} \end{pmatrix} = \begin{pmatrix} r(0) & r(1) & \dots & r(P-1) & r(T+Q-1) & r(T+Q) & \dots & r(T-Q-1) \\ r(1) & r(0) & \dots & r(P-2) & r(T+Q-2) & r(T+Q-1) & \dots & r(T+Q-2) \\ r(2) & r(1) & \dots & r(P-3) & r(T+Q-3) & r(T+Q-2) & \dots & r(T+Q-3) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(P-1) & r(P-2) & \dots & r(0) & r(T+Q-P) & r(T+Q-P+1) & \dots & r(T+Q-P) \\ r(T+Q-1) & r(T+Q-2) & \dots & r(T+Q-P) & r(0) & r(1) & \dots & r(2Q) \\ r(T+Q) & r(T+Q-1) & \dots & r(T+Q-P+1) & r(1) & r(0) & \dots & r(2Q-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ r(T-Q-1) & r(T-Q-2) & \dots & r(T-Q-P) & r(2Q) & r(2Q-1) & \dots & r(0) \end{pmatrix}^{-1} \begin{pmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(P) \\ r(T+Q) \\ r(T+Q-1) \\ \vdots \\ r(T-Q) \end{pmatrix}$$

(10.66)

The modified AR model can be used for interpolation in the same way as the conventional AR model described in the previous section. Again, it is assumed that within a data window of *N* speech samples, a segment of *M* samples commencing from the sample point *k*, $\boldsymbol{x}_{\mathrm{Uk}}=\{x(k),\ x(k+1),\ \ldots,$

$x(k+M-1)\}$ is missing. Figure 10.11 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 (10.64) can be written in matrix form as

$$\begin{pmatrix} e(T+Q) \\ e(T+Q+1) \\ \vdots \\ e(k-1) \\ \hline e(k) \\ e(k+1) \\ e(k+2) \\ \vdots \\ e(k+M+P-2) \\ e(k+M+P-1) \\ \hline e(k+M+P) \\ e(k+M+P+1) \\ \vdots \\ e(N-1) \end{pmatrix} = \begin{pmatrix} x(T+Q) \\ x(T+Q+1) \\ \vdots \\ x(k-1) \\ \hline x_{UK}(k) \\ x_{UK}(k+1) \\ x_{UK}(k+2) \\ \vdots \\ x(k+M+P-2) \\ x(k+M+P-1) \\ \hline x(k+M+P) \\ x(k+M+P+1) \\ \vdots \\ x(N-1) \end{pmatrix} - \begin{pmatrix} x(T+Q-1) & \cdots & x(T+Q-P) & x(2Q) & \cdots & x(0) \\ x(T+Q) & \cdots & x(T+Q-P+1) & x(2Q+1) & \cdots & x(1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x(k-2) & \cdots & x(k-P-1) & x(k-T+Q-1) & \cdots & x(k-T-Q-1) \\ \hline x(k-1) & \cdots & x(k-P) & x(k-T+Q) & \cdots & x(k-T-Q) \\ x_{UK}(k) & \cdots & x(k-P+1) & x(k-T+Q+1) & \cdots & x(k-T-Q+1) \\ x_{UK}(k+1) & \cdots & x(k-P+2) & x(k-T+Q+2) & \cdots & x(k-T-Q+2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x(k+M+P-3) & \cdots & x_{UK}(k+M-2) & x(k+M+P-T+Q-2) & \cdots & x(k+M+P-T-Q-2) \\ x(k+M+P-2) & \cdots & x_{UK}(k+M-1) & x(k+M+P-T+Q-1) & \cdots & x(k+M+P-T-Q-1) \\ \hline x(k+M+P-1) & \cdots & x(k+M) & x(k+M+P-T+Q) & \cdots & x(k+M+P-T-Q) \\ x(k+M+P) & \cdots & x(k+M+1) & x(k+M+P-T+Q+1) & \cdots & x(k+M+P-T-Q+1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x(N-2) & \cdots & x(N-P-1) & x(N-T+Q-1) & \cdots & x(N-T-Q-1) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \\ p_{-Q} \\ \vdots \\ p_{+Q} \end{pmatrix} \tag{10.67}$$

where the subscript Uk denotes the unknown samples. In compact matrix notation, this set of equation can be written in the form

$$\boldsymbol{e}(\boldsymbol{x}_{Uk}, \boldsymbol{c}) = \boldsymbol{x} + \boldsymbol{X}\boldsymbol{c} \tag{10.68}$$

As in Section 10.3.2, the interpolation problem is solved in two stages:

(a) In the first stage, the known samples on both sides of the missing signal are used to estimate the AR coefficient vector $\boldsymbol{c}$.
(b) In the second stage, the AR coefficient estimates are substituted in Equation (10.68) so that the only unknowns are the data samples.

The solution follows the same steps as those described in Section 10.3.2.

10.3.5 LSAR Interpolation Error

In this section, we discuss the effects of the signal characteristics, the model parameters and the number of unknown samples on the interpolation error. The interpolation error $v(m)$, defined as the difference between the original sample $x(m)$ and the interpolated sample $\hat{x}(m)$, is given by

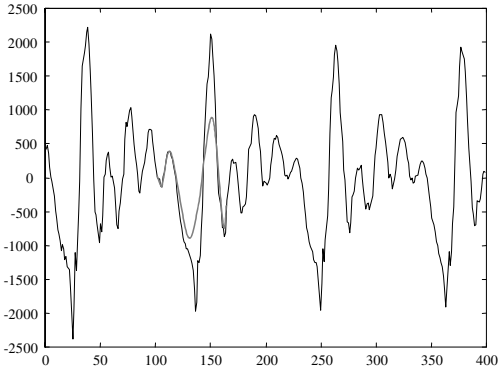
$$v(m) = x(m) - \hat{x}(m) \tag{10.69}$$

A common measure of signal distortion is the mean square error distance defined as

$$D(\mathbf{c}, M) = \frac{1}{M} \mathcal{E}\left\{ \sum_{m=0}^{M-1} [x(k+m) - \hat{x}(k+m)]^2 \right\} \tag{10.70}$$

where k is the beginning of an M-samples long segment of missing signal, and $\mathcal{E}$ [.] is the expectation operator. In Equation (10.70), the average distortion D is expressed as a function of the number of the unknown samples M, and also the model coefficient vector $\boldsymbol{c}$. In general, the quality of interpolation depends on the following factors:

(a) *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.
(b) *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 (10.63) becomes increasingly ill-conditioned as the length of the missing samples increases.
(c) *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.
(d) *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.



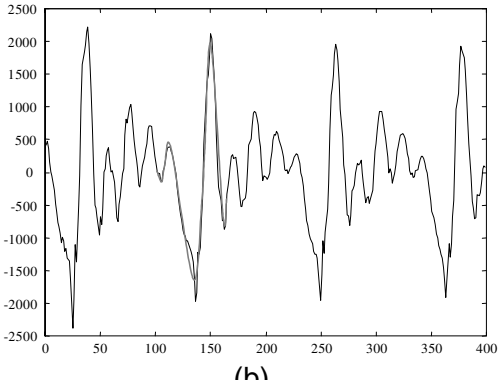
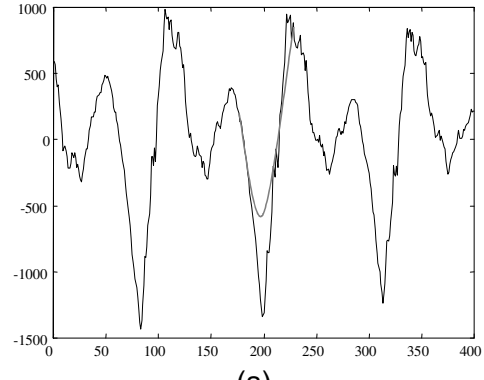


Figure 10.12 (a) A section of speech showing interpolation of 60 samples starting from the sample point 100 (b) Interpolation using short and long-term correlations. Interpolated samples are shown by the light shaded line.



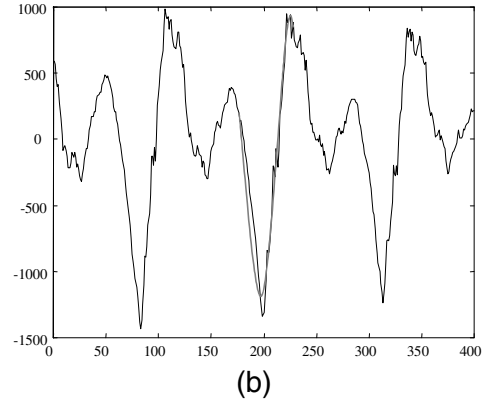


Figure 10.13 (a) A section of speech showing interpolation of 50 samples starting from the sample point 175 (b) Interpolation using short and long-term correlations. Interpolated samples are shown by the light shaded line.

The interpolation error also depends on how well the AR parameters can be estimated from the incomplete data. In Equation (10.54), in the first stage of the solution, where the AR coefficients are estimated, two different approaches may be employed to linearise the system of equations. In the first approach all equations, between the dashed lines, that involve nonlinear terms are discarded. This approach has the advantage that no assumption is made about the missing samples. In fact, from a signal-ensemble point of view, the effect of discarding some equations is

equivalent to that of having a smaller signal record. In the second method, starting from an initial estimate of the unknown vector (such as $\boldsymbol{x}_{Uk}=\mathbf{0}$), Equation (10.54) is solved to obtain the AR parameters. The AR coefficients are then used in the second stage of the algorithm to estimate the unknown samples. These estimates may be improved in further iterations of the algorithm. The algorithm usually converges after one or two iterations.

Figures 10.12 and 10.13 show the results of application of the least square error AR interpolation method to speech signals. The interpolated speech segments were chosen to coincide with the onset of an excitation pulse. In these experimental cases the original signals are available for comparison. Each signal was interpolated by the AR model of Equation (10.53) and also by the extended AR model of Equation (10.64). The length of the conventional linear predictor model was set to 20. The modified linear AR model of Equation (10.64) has a prediction order of (20,7); that is, the short-term predictor has 20 coefficients and the long-term predictor has 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.

10.3.6 Interpolation in Frequency–Time Domain

Time-domain, AR model-based interpolation methods are effective for the interpolation of a relatively short length of samples (say less than 100 samples at a 20 kHz sampling rate), but suffer severe performance degradations when used for interpolation of large sequence of samples. This is partly due to the numerical problems associated with the inversion of a large matrix, involved in the time-domain interpolation of a large number of samples, Equation (10.58).

Spectral–time representation provides a useful form for the interpolation of a large gap of missing samples. For example, through discrete Fourier transformation (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.

Spectral–Time Representation with STFT

A relatively simple and practical method for spectral–time representation of a signal is the short-time Fourier transform (STFT) method. To construct a

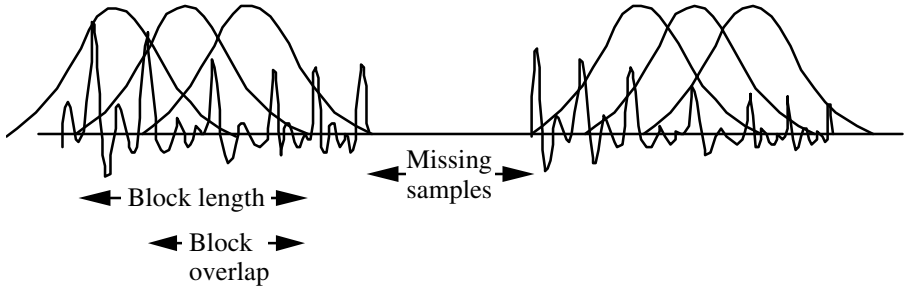


Figure 10.14 Illustration of segmentation of a signal (with a missing gap) for spectral-time representation.

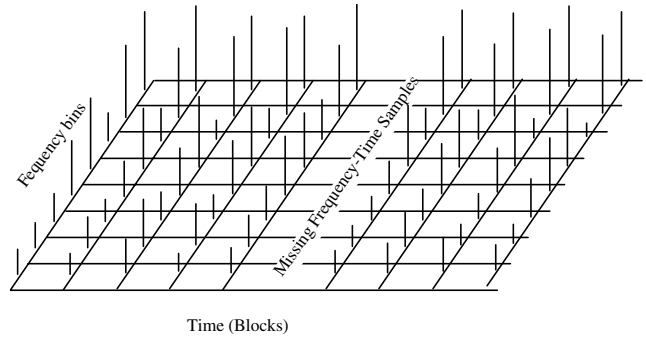


Figure 10.15 Spectral-time representation of a signal with a missing gap.

two-dimensional STFT from a one-dimensional function of time $x(m)$, the input signal is segmented into overlapping blocks of N samples, as illustrated in Figure 10.14. 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

$$X(k,m)=\sum_{i=0}^{N-1} w(i)\,x\big(m(N-D)+i\big)e^{-j\frac{2\pi}{N}ik}, \quad k=0,\ ...,\ N-1 \quad (10.71)$$

where $X(k,m)$ is a spectral–time representation with time index m and 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

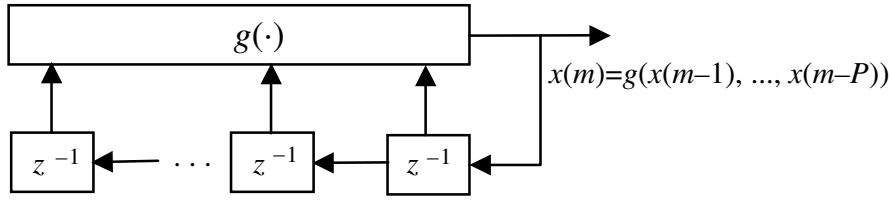


Figure 10.16 Configuration of a digital oscillator.

the blocks. In general, the k^{th} spectral component of a signal has a time-varying character, i.e. it is "born", evolves for some time, disappears, and then reappears with a different intensity and a different characteristics. Figure 10.15 illustrates a spectralthtime 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.

10.3.7 Interpolation Using Adaptive Code Books

In the LSAR interpolation method, described in Section 10.3.2, the signals are modelled as the output of an AR model excited by a random input. Given enough samples, the AR coefficients can be estimated with reasonable accuracy. However, the instantaneous values of the random excitation during the periods when the signal is missing cannot be recovered. This leads to a consistent underestimation of the amplitude and the energy of the interpolated samples. One solution to this problem is to use a zero-input signal model. Zero-input models are feedback oscillator systems that produce an output signal without requiring an input.

The general form of the equation describing a digital nonlinear oscillator can be expressed as

$$x(m)=g_f\left(x(m-1),x(m-2),\ldots,x(m-P)\right) \tag{10.72}$$

The mapping function $g_f(\cdot)$ may be a parametric or a non-parametric mapping. The model in Equation (10.72) can be considered as a nonlinear

predictor, and the subscript *f* denotes forward prediction based on the past samples.

A parametric model of a nonlinear oscillator can be formulated using a Volterra filter model. However, in this section, we consider a 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 $(P+1)^{th}$ sample is intended as an output. Given P input samples $\boldsymbol{x}=[x(m-1), ..., x(m-P)]$, the codebook output is the $(P+1)^{th}$ sample of the vector in the codebook whose first P samples have a minimum distance from the input signal $\boldsymbol{x}$. 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

$$x_b(m)=g_b\left(x(m+1),x(m+2),\cdots,x(m+P)\right) \tag{10.73}$$

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 $\boldsymbol{x}=[x(m), ..., x(m+P-1)]$ 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 ouputs of the forward and backward nonlinear oscillators may be combined as

$$\hat{x}(k+m)=\left(\frac{M-1-m}{M-1}\right)\hat{x}_f(k+m)+\left(\frac{m}{M-1}\right)\hat{x}_b(k+m) \tag{10.74}$$

.

where it is assumed that the missing samples start at k.

10.3.8 Interpolation Through Signal Substitution

Audio signals often have a time-varying but 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).

10.4 Summary

Interpolators, in their various forms, are used in most signal processing applications. The obvious example is the estimation of a sequence of missing samples. However, the use of an interpolator covers a much wider range of applications, from low-bit-rate speech coding to pattern recognition and decision making systems. We started this chapter with a study of the ideal interpolation of a band-limited signal, and its applications in digital-to-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, pp. 180–183, Glasgow.

COHEN L. (1989) Time-Frequency Distributions-A review. Proc. IEEE, **77(7)**, pp. 941–81.

CROCHIERE R.E. and RABINER L.R. (1981) Interpolation and Decimation of Digital Signals-A Tutorial review. Proc. IEEE, **69**, pp. 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**, pp. 317–330 June.

KAY S.M. (1983) Some Results in Linear Interpolation Theory. IEEE Trans. Acoustics Speech and Signal Processing, **ASSP-31**, pp. 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**, pp. 2043–2045.

KUBIN G. and KLEIJN W.B. (1994) Time-Scale Modification of Speech Based On A Nonlinear Oscillator Model. Proc. IEEE Int. Conf., Speech and Signal Processing, ICASSP-94, pp. 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, pp. 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**. pp. 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 Ed. 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**, pp. 692–702, June.

STEELE R. and JAYANT N.S. (1980) Statistical Block Coding for DPCM-AQF Speech. IEEE Trans on Communications. **COM-28**, **11**, pp. 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.

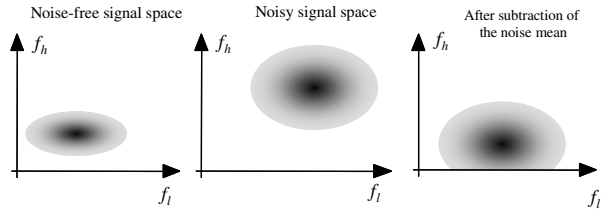
VERHELST W. and ROELANDS M. (1993) An Overlap-Add Technique Based on Waveform Similarity (Wsola) for High Quality Time-Scale Modification of Speech. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP-93, pp. II-554-II-557, Adelaide.

WIENER N. (1949) Extrapolation, Interpolation and Smoothing of Stationary Time Series With Engineering Applications. MIT Press, Cambridge, MA.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

11



SPECTRAL SUBTRACTION

Spectral subtraction 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 and only the noise is present. The assumption is that the noise is a stationary or a slowly varying process, and that the noise spectrum does not change significantly in-between the update periods. 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 relatively 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. The processing distortion becomes more noticeable as the signal-to-noise ratio decreases. In this chapter, we study spectral subtraction, and the different methods of reducing and removing the processing distortions.

11.1 Spectral Subtraction

In applications where, in addition to the noisy signal, the noise is accessible on a separate channel, it may be possible to retrieve the signal by subtracting an estimate of the noise from the noisy signal. For example, the adaptive noise canceller of Section 1.3.1 takes as the inputs the noise and the noisy signal, and outputs an estimate of the clean signal. However, in many applications, such as at the receiver of a noisy communication channel, the only signal that is available is the noisy signal. In these situations, 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 signal spectrum. The effect of additive noise on the magnitude spectrum of a signal is to increase the mean and the variance of the spectrum as illustrated in Figure 11.1. The increase in the variance of the signal spectrum results from the random fluctuations of the noise, and cannot be cancelled out. The increase in the mean of the signal spectrum can be removed by subtraction of an estimate of the mean of the noise spectrum from the noisy signal spectrum. The noisy signal model in the time domain is given by

$$y(m)=x(m)+n(m) \tag{11.1}$$

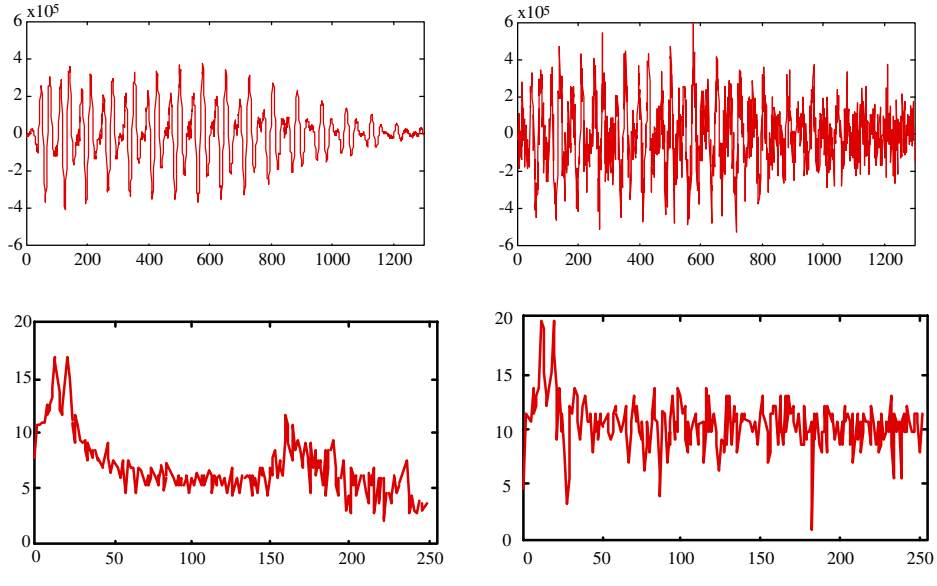


Figure 11.1 Illustrations of the effect of noise on a signal in the time and the frequency domains.

where $y(m)$, $x(m)$ and $n(m)$ are the signal, the additive noise and the noisy signal respectively, and m is the discrete time index. In the frequency domain, the noisy signal model of Equation (11.1) is expressed as

$$Y(f)=X(f)+N(f) \tag{11.2}$$

where $Y(f)$, $X(f)$ and $N(f)$ are the Fourier transforms of the noisy signal $y(m)$, the original signal $x(m)$ and the noise $n(m)$ respectively, and f is the frequency variable. In spectral subtraction, the incoming signal $x(m)$ is buffered and divided into segments of N samples length. Each segment is windowed, using a Hanning or a Hamming window, 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

$$\begin{aligned} y_w(m) &= w(m)y(m) \\ &= w(m)[x(m)+n(m)] \\ &= x_w(m)+n_w(m) \end{aligned} \tag{11.3}$$

The windowing operation can be expressed in the frequency domain as

$$\begin{aligned} Y_w(f) &= W(f)*Y(f) \\ &= X_w(f)+N_w(f) \end{aligned} \tag{11.4}$$

where the operator * denotes convolution. 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.
Figure 11.2 illustrates a block diagram configuration of the spectral subtraction method. A more detailed implementation is described in Section 11.4. The equation describing spectral subtraction may be expressed as

$$\left|\hat{X}(f)\right|^b = |Y(f)|^b - \alpha\overline{|N(f)|^b} \tag{11.5}$$

where $|\hat{X}(f)|^b$ is an estimate of the original signal spectrum $|X(f)|^b$ and $\overline{|N(f)|^b}$ is the time-averaged noise spectra. It is assumed that the noise is a wide-sense stationary random process. For magnitude spectral subtraction, the exponent b=1, and for power spectral subtraction, b=2. The parameter α

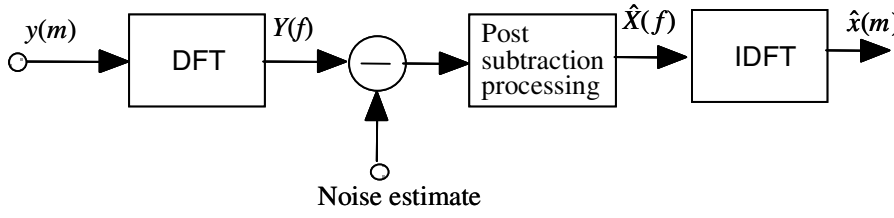


Figure 11.2 A block diagram illustration of spectral subtraction.

in Equation (11.5) controls the amount of noise subtracted from the noisy signal. For full noise subtraction, α=1 and for over-subtraction α>1. The time-averaged noise spectrum is obtained from the periods when the signal is absent and only the noise is present as

$$\overline{|N(f)|^b} = \frac{1}{K}\sum_{i=0}^{K-1}|N_i(f)|^b \tag{11.6}$$

In Equation (11.6), $|N_i(f)|$ is the spectrum of the i^{th} noise frame, and it is assumed that there are K frames in a noise-only period, where K is a variable. Alternatively, the averaged noise spectrum can be obtained as the output of a first order digital low-pass filter as

$$\overline{|N_i(f)|^b} = \rho\,\overline{|N_{i-1}(f)|^b} + (1-\rho)|N_i(f)|^b \tag{11.7}$$

where the low-pass filter coefficient ρ is typically set between 0.85 and 0.99. For restoration of a time-domain signal, the magnitude spectrum estimate $|\hat{X}(f)|$ is combined with the phase of the noisy signal, and then transformed into the time domain via the inverse discrete Fourier transform as

$$\hat{x}(m) = \sum_{k=0}^{N-1}|\hat{X}(k)|e^{j\theta_Y(k)}e^{-j\frac{2\pi}{N}km} \tag{11.8}$$

where $\theta_Y(k)$ is the phase of the noisy signal frequency $Y(k)$. The signal restoration equation (11.8) 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 magnitude 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

$$T[|\hat{X}(f)|]=\begin{cases}|\hat{X}(f)| & \text{if } |\hat{X}(f)|>\beta|Y(f)| \\ \text{fn}[|Y(f)|] & \text{otherwise}\end{cases} \tag{11.9}$$

For example, we may chose a rule such that if the estimate $|\hat{X}(f)|>0.01|Y(f)|$ (in magnitude spectrum 0.01 is equivalent to –40 dB) then $|\hat{X}(f)|$ should be set to some function of the noisy signal fn[$Y(f)$]. In its simplest form, fn[$Y(f)$]=*noise floor*, where the noise floor is a positive constant. An alternative choice is fn[$|Y(f)|$]=$\beta|Y(f)|$. In this case,

$$T[|\hat{X}(f)|]=\begin{cases}|\hat{X}(f)| & \text{if } |\hat{X}(f)|>\beta|Y(f)| \\ \beta|Y(f)| & \text{otherwise}\end{cases} \tag{11.10}$$

Spectral subtraction may be implemented in the power or the magnitude spectral domains. The two methods are similar, although theoretically they result in somewhat different expected performance.

11.1.1 Power Spectrum Subtraction

The power spectrum subtraction, or squared-magnitude spectrum subtraction, is defined by the following equation:

$$|\hat{X}(f)|^2=|Y(f)|^2-\overline{|N(f)|^2} \tag{11.11}$$

where it is assumed that α, the subtraction factor in Equation (11.5), is unity. We denote the power spectrum by $\mathcal{E}[|X(f)|^2]$, the time-averaged power spectrum by $\overline{|X(f)|^2}$ and the *instantaneous* power spectrum by $|X(f)|^2$. By expanding the instantaneous power spectrum of the noisy

signal $|Y(f)|^2$, and grouping the appropriate terms, Equation (11.11) may be rewritten as

$$|\hat{X}(f)|^2 = |X(f)|^2 + \underbrace{\left(|N(f)|^2 - \overline{|N(f)|^2}\right)}_{\text{Noise variations}} + \underbrace{X^*(f)N(f) + X(f)N^*(f)}_{\text{Cross products}} \tag{11.12}$$

Taking the expectations of both sides of Equation (11.12), and assuming that the signal and the noise are uncorrelated ergodic processes, we have

$$\mathcal{E}[|\hat{X}(f)|^2] = \mathcal{E}[|X(f)|^2] \tag{11.13}$$

From Equation (11.13), 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 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 (11.13), we have not considered non-linear rectification of the negative estimates of the squared magnitude spectrum.

11.1.2 Magnitude Spectrum Subtraction

The magnitude spectrum subtraction is defined as

$$|\hat{X}(f)| = |Y(f)| - \overline{|N(f)|} \tag{11.14}$$

where $\overline{|N(f)|}$ is the time-averaged magnitude spectrum of the noise. Taking the expectation of Equation (11.14), we have

$$\begin{aligned} \mathcal{E}[|\hat{X}(f)|] &= \mathcal{E}[|Y(f)|] - \mathcal{E}[\overline{|N(f)|}] \\ &= \mathcal{E}[|X(f) + N(f)|] - \mathcal{E}[\overline{|N(f)|}] \\ &\approx \mathcal{E}[|X(f)|] \end{aligned} \tag{11.15}$$

For signal restoration the magnitude estimate is combined with the phase of the noisy signal and then transformed into the time domain using Equation (11.8).

11.1.3 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

$$\begin{aligned} |\hat{X}(f)|^2 &= |Y(f)|^2 - \overline{|N(f)|^2} \\ &= H(f)|Y(f)|^2 \end{aligned} \tag{11.16}$$

where $H(f)$, the frequency response of the spectral subtraction filter, is defined as

$$\begin{aligned} H(f) &= 1 - \frac{\overline{|N(f)|^2}}{|Y(f)|^2} \\ &= \frac{|Y(f)|^2 - \overline{|N(f)|^2}}{|Y(f)|^2} \end{aligned} \tag{11.17}$$

The spectral subtraction filter $H(f)$ is a zero-phase filter, with its magnitude response in the range $0 \geq H(f) \geq 1$. 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

$$W(f) = \frac{\mathcal{E}[|Y(f)|^2] - \mathcal{E}[|N(f)|^2]}{\mathcal{E}[|Y(f)|^2]} \tag{11.18}$$

A comparison of $W(f)$ and $H(f)$, from Equations (11.18) and (11.17), 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 process. However, assuming that the signal and noise are wide-sense stationary ergodic processes, we may replace the instantaneous noisy signal spectrum $|Y(f)|^2$ in the spectral subtraction equation (11.18) with the time-averaged spectrum $\overline{|Y(f)|^2}$, to obtain

$$H(f) = \frac{\overline{|Y(f)|^2} - \overline{|N(f)|^2}}{\overline{|Y(f)|^2}} \tag{11.19}$$

For an ergodic process, as the length of the time over which the signals are averaged increases, the time-averaged spectrum approaches the ensemble-averaged spectrum, and in the limit, the spectral subtraction filter of Equation (11.19) approaches the Wiener filter equation (11.18). 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.

11.2 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 (11.12) and the constraint that the magnitude spectrum must have a non-negative value, we may identify three sources of distortions of the instantaneous estimate of the magnitude or power spectrum as:

(a) the variations of the instantaneous noise power spectrum about the mean;
(b) the signal and noise cross-product terms;
(c) 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

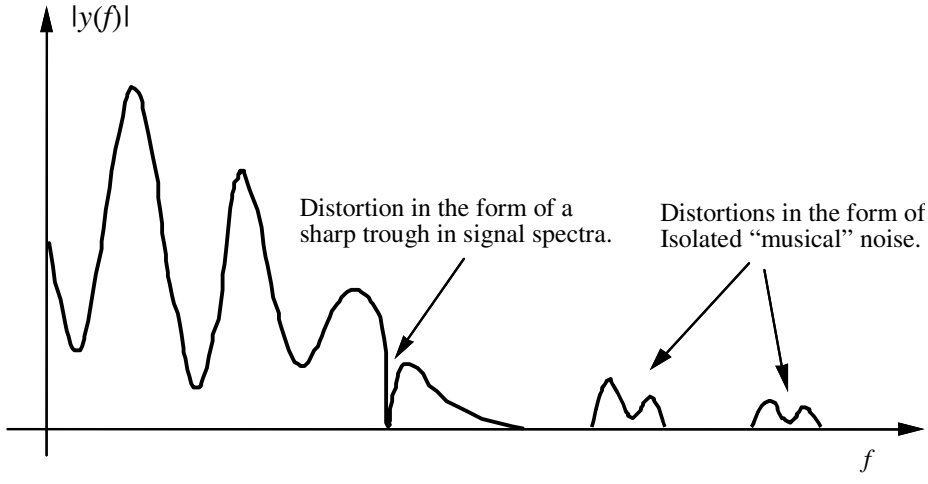


Figure 11.3 Illustration of distortions that may result from spectral subtraction.

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:

(a) a sharp trough or peak in the signal spectra;
(b) isolated narrow bands of frequencies.

In the vicinity of a high amplitude signal frequency, the noise-induced trough or peak is often masked, and made inaudible, by the high signal energy. The main cause of audible degradations is the isolated frequency components also known as *musical tones* or musical noise illustrated in Figure 11.3. 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.

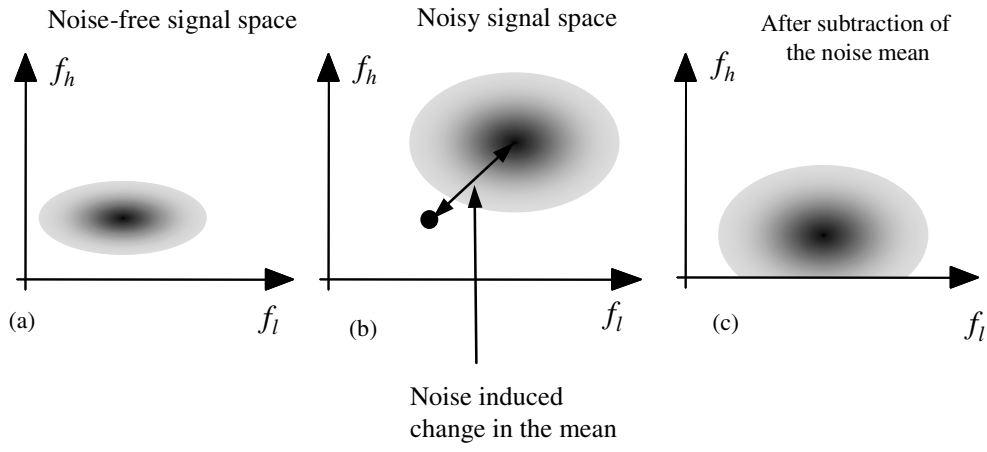


Figure 11.4 Illustration of the distorting effect of spectral subtraction on the space of the magnitude spectrum of a signal.

11.2.1 Effect of Spectral Subtraction on Signal Distribution

Figure 11.4 is an illustration of the distorting effect of spectral subtraction on the distribution of the magnitude spectrum of a signal. In this figure, we have considered the simple case where the spectrum of a signal is divided into two parts; a low-frequency band f_l and a high-frequency band f_h. Each point in Figure 11.4 is a plot of the high-frequency spectrum versus the low-frequency spectrum, in a two-dimensional signal space. Figure 11.4(a) shows an assumed distribution of the spectral samples of a signal in the two-dimensional magnitude–frequency space. The effect of the random noise, shown in Figure 11.4(b), is an increase in the mean and the variance of the spectrum, by an amount that depends on the mean and the variance of the magnitude spectrum of the noise. The increase in the variance constitutes an irrevocable distortion. The increase in the mean of the magnitude spectrum can be removed through spectral subtraction. Figure 11.4(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.

11.2.2 Reducing the Noise Variance

The distortions that result from spectral subtraction are due to the variations of the noise spectrum. In Section 9.2 we considered the methods of reducing the 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 $N(f)$ is given by

$$\mathrm{Var}[|N(f)|^2] \approx P_{NN}^2(f) = \sigma_n^4 \tag{11.20}$$

and the variance of the running average of K independent spectral components is

$$\mathrm{Var}\left[\frac{1}{K}\sum_{i=0}^{K-1}|N_i(f)|^2\right] \approx \frac{1}{K}P_{NN}^2(f) \approx \frac{1}{K}\sigma_n^4 \tag{11.21}$$

From Equation (11.21), 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 blocks of N samples. Each signal block is then transformed via a DFT into a block of N spectral samples $Y(f)$. Successive blocks of spectral samples form a two-dimensional frequency–time matrix denoted by $Y(f,t)$ where the variable t is the segment index and denotes the time dimension. The signal $Y(f,t)$ can be considered as a band-pass channel f that contains a time-varying signal $X(f,t)$ plus a random noise component $N(f,t)$. 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

$$|Y_{LP}(f,t)| = \rho|Y_{LP}(f,t-1)| + (1-\rho)|Y(f,t)| \tag{11.22}$$

where the subscript LP denotes the output of the low-pass filter, and the smoothing coefficient ρ controls the bandwidth and the time constant of the low-pass filter.

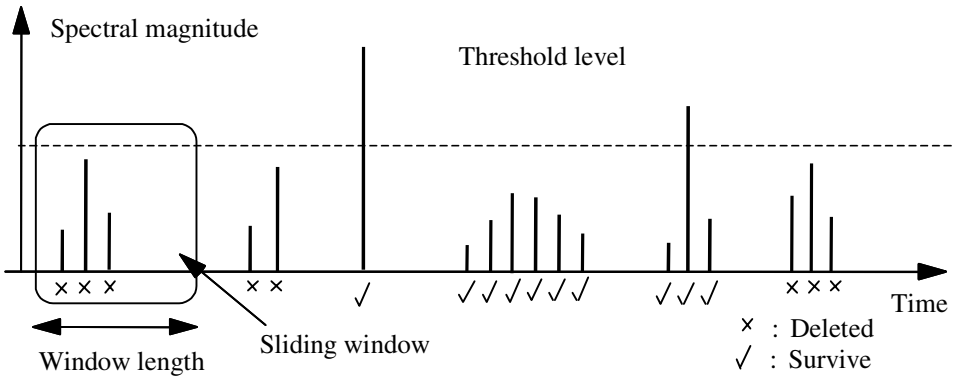


Figure 11.5 Illustration of a method for identification and filtering of "musical noise".

11.2.3 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 intensity and spectral composition. The time–varying nature of audio signals plays an important role in conveying information, sensation and quality. The musical tone noise, introduced as an undesirable by-product of spectral subtraction, is also time-varying. However, there are significant differences between the characteristics of most audio signals and so-called musical noise. The characteristic differences may be used to identify and remove some of the more annoying distortions. Identification of musical noise may be achieved by examining the variations of the signal in the time and frequency domains. The main characteristics of musical noise are that it tends to be relatively short-lived random isolated bursts of narrow band signals, with relatively small amplitudes.

Using a DFT block size of 128 samples, at a sampling rate of 20 kHz, experiments indicate that the great majority of musical noise tends to last no more than three frames, whereas genuine signal frequencies have a considerably longer duration. This observation was used as the basis of an effective "musical noise" suppression system. Figure 11.5 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.

11.3 Non-Linear Spectral Subtraction

The use of spectral subtraction in its basic form of Equation (11.5) 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 a number of 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:

$$\begin{aligned} |\hat{X}(f)| &= |Y(f)| - \overline{|N(f)|} \\ &\approx |X(f)| + |N(f)| - \overline{|N(f)|} \\ &\approx |X(f)| + V_N(f) \end{aligned} \tag{11.23}$$

where $V_N(f)$ is the zero-mean random component of the noise spectrum. If $V_N(f)$ is well above the signal $X(f)$ 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:

$$|\hat{X}(f)| = Y(f) - \alpha\left(SNR(f)\right)\overline{|N(f)|}_{NL} \tag{11.24}$$

where $\alpha(SNR(f))$ is an SNR-dependent subtraction factor and $\overline{|N(f)|}_{NL}$ is a non-linear estimate of the noise spectrum. The spectral estimate is further processed to avoid negative estimates as

$$|\hat{X}(f)| = \begin{cases} |\hat{X}(f)| & \text{if } |\hat{X}(f)| > |\beta Y(f)| \\ |\beta Y(f)| & \text{otherwise} \end{cases} \tag{11.25}$$

One form of an SNR-dependent subtraction factor for Equation (11.24) is given by

$$\alpha\left(SNR(f)\right) = 1 + \frac{sd\left(|N(f)|\right)}{\overline{|N(f)|}} \tag{11.26}$$

where the function $sd(|N(f)|$ is the standard deviation of the noise at frequency f. For white noise, $sd(|N(f)|=\sigma_n$, where σ_n^2 is the noise variance. Substitution of Equation (11.26) in Equation (11.24) yields

$$|\hat{X}(f)| = |Y(f)| - \left[1 + \frac{sd\left(|N(f)|\right)}{|N(f)|}\right]\overline{|N(f)|} \tag{11.27}$$

In Equation (11.27) 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, $\alpha(SNR(f))=1$, and at the other extreme for white noise $\alpha(SNR(f))=2$. In application of spectral subtraction to speech recognition, it is found that the best subtraction factor is usually between 1 and 2.

In the non-linear spectral subtraction method of Lockwood and Boudy, the spectral subtraction filter is obtained from

$$H(f) = \frac{\overline{|Y(f)|^2} - \overline{|N(f)|_{NL}^2}}{\overline{|Y(f)|^2}} \tag{11.28}$$

Lockwood and Boudy suggested the following function as a non-linear estimator of the noise spectrum:

$$\overline{|N(f)|^2}_{NL} = \Phi\left(\max_{\text{over } M \text{ frames}}\left(|N(f)|^2\right), SNR(f), \overline{|N(f)|^2}\right) \tag{11.29}$$

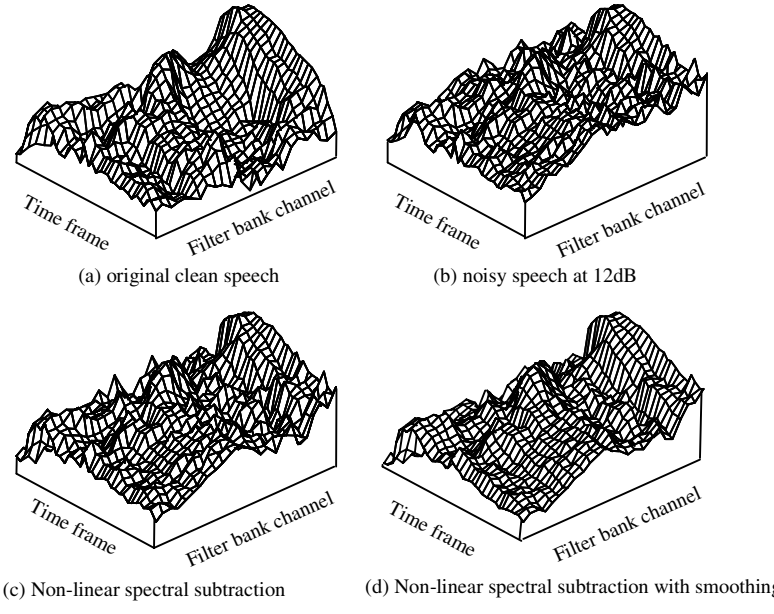


Figure 11.6 Illustration of the effects of non-linear spectral subtraction.

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 $\Phi(\cdot)$ is given by the following equation:

$$\Phi\left(\max_{\text{over } M \text{ frames}}\left(|N(f)|^2\right), SNR(f)\right) = \frac{\max_{\text{Over } M \text{ frames}}\left(|N(f)|^2\right)}{1+\gamma\, SNR(f)} \tag{11.30}$$

where γ is a design parameter. From Equation (11.30) as the SNR decreases the output of the non-linear estimator $\Phi(\cdot)$ approaches $\max(|N(f)|^2)$, 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:

$$\overline{|N(f)|^2} \le \Phi\left(\max_{\text{over } M \text{ frames}}\left(|N(f)|^2\right), SNR(f), \overline{|N(f)|^2}\right) \le 3\overline{|N(f)|^2} \tag{11.31}$$

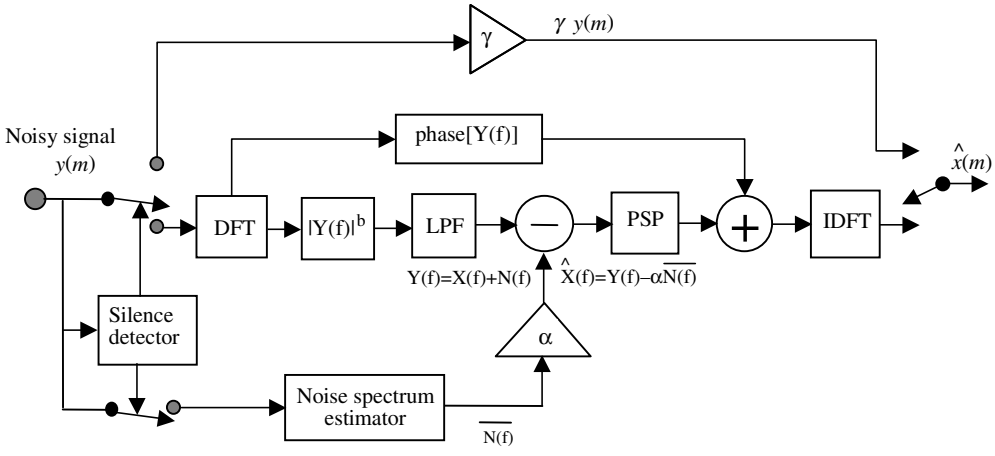


Figure 11.7 Block diagram configuration of a spectral subtraction system. PSP = post spectral subtraction processing.

The maximum attenuation of the spectral subtraction filter is limited to $H(f) \geq \beta$, where usually the lower bound $\beta \geq 0.01$. Figure 11.6 illustrates the effects of non-linear spectral subtraction and smoothing in restoration of the spectrum of a speech signal.

11.4 Implementation of Spectral Subtraction

Figure 11.7 is a block diagram illustration of a spectral subtraction system. It includes the following subsystems:

(a) a silence detector for detection of the periods of signal inactivity; the noise spectra is updated during these periods;
(b) a discrete Fourier transformer (DFT) for transforming the time domain signal to the frequency domain; the DFT is followed by a magnitude operator;
(c) a lowpass filter (LPF) for reducing the noise variance; the purpose of the LPF is to reduce the processing distortions due to noise variations;
(d) a post-processor for removing the processing distortions introduced by spectral subtraction.;
(e) an inverse discrete Fourier transform (IDFT) for transforming the processed signal to the time domain.
(f) an attenuator γ for attenuation of the noise during silent periods.

The DFT-based spectral subtraction is a block processing algorithm. The incoming audio signal is buffered and divided into overlapping blocks of N samples as shown in Figure 11.7. Each block is Hanning (or Hamming) 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 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 11.8) 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

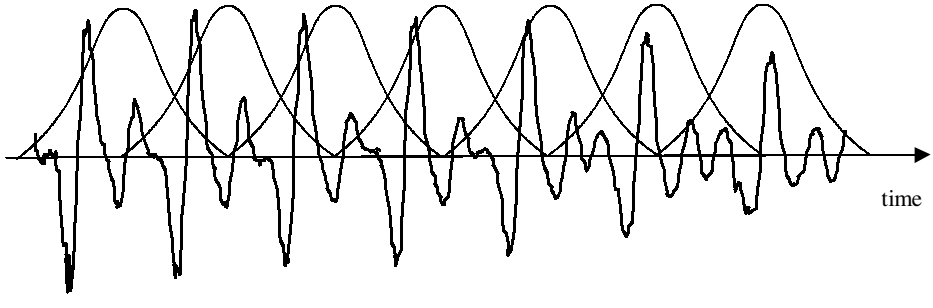


Figure 11.8 Illustration of the window and overlap process in spectral subtraction.

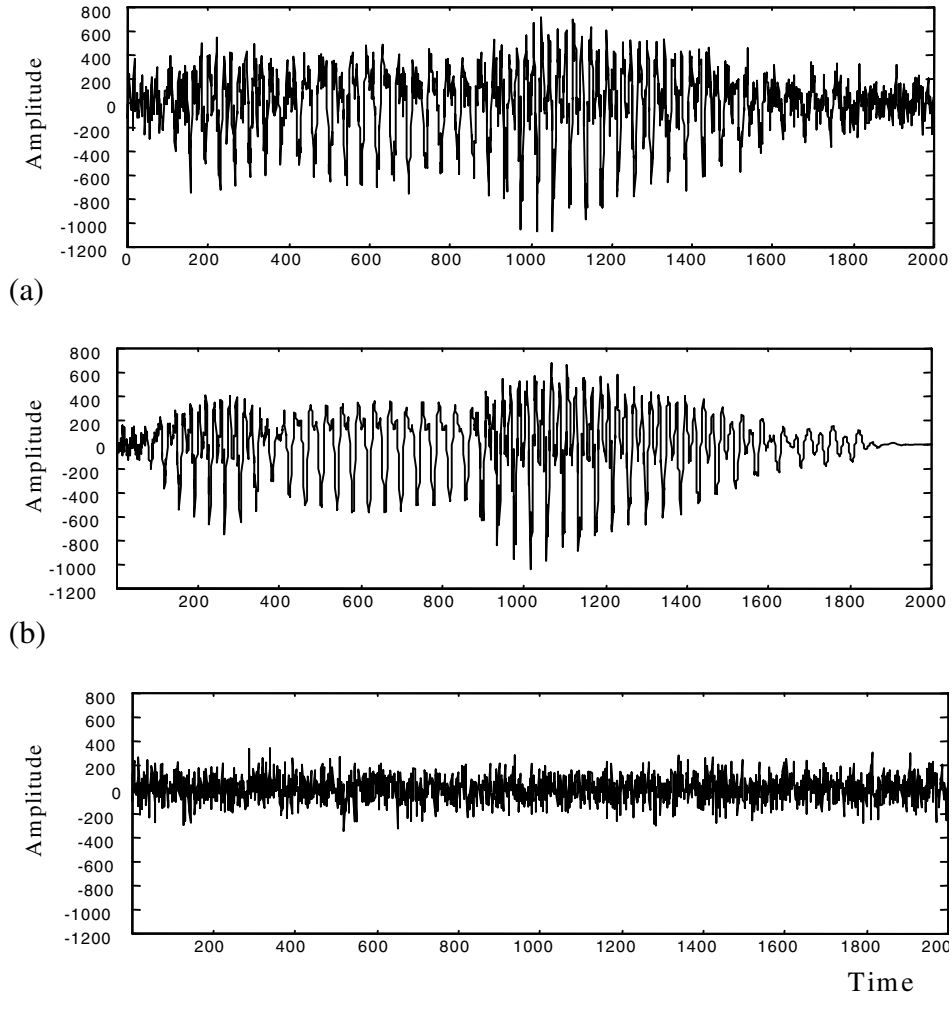


Figure 11.9 (a) A noisy signal. (b) Restored signal after spectral subtraction. (c) Noise estimate obtained by subtracting (b) from (a).

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.

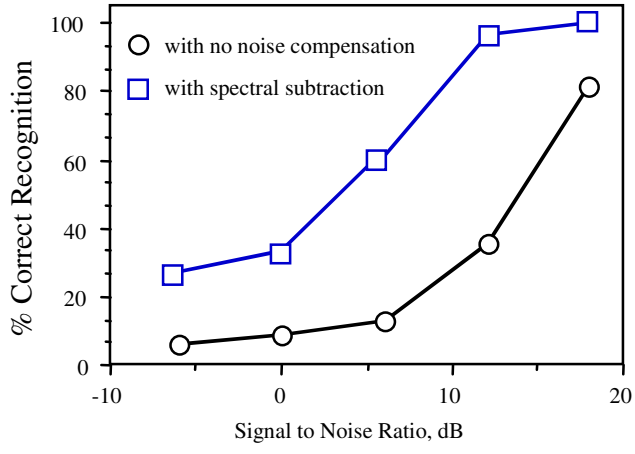


Figure 11.10 The effect of spectral subtraction in improving speech recognition (for a spoken digit data base) in the presence of helicopter noise.

11.4.1 Application to Speech Restoration and Recognition

In speech restoration, the objective is to estimate the instantaneous signal spectrum $X(f)$. The restored magnitude spectrum is combined with the phase of the noisy signal to form the restored speech signal. In contrast, speech recognition systems are more concerned with the restoration of the envelope of the short-time spectrum than the detailed structure of the spectrum. Averaged values, such as the envelope of a spectrum, can often be estimated with more accuracy than the instantaneous values. However, in speech recognition, as in signal restoration, the processing distortion due to the negative spectral estimates can cause substantial deterioration in performance. A careful implementation of spectral subtraction can result in a significant improvement in the recognition performance.

Figure 11.9 illustrates the effects of spectral subtraction in restoring a section of a speech signal contaminated with white noise. Figure 11.10 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.

11.5 Summary

This chapter began with an introduction to spectral subtraction and its relation to Wiener filters. The main attraction of spectral subtraction is its relative simplicity, in that it only requires an estimate of the noise power spectrum. However, this can also be viewed as a fundamental limitation in that spectral subtraction does not utilise the statistics and the distributions of the signal process. The main problem in spectral subtraction is the presence of processing distortions caused by the random variations of the noise. The estimates of the magnitude and power spectral variables, that owing to noise variations, are negative, have to be mapped into non-negative values. In Section 11.2, we considered the processing distortions, and illustrated the effects of rectification of negative estimates on the distribution of the signal spectrum. In Section 11.3, a number of 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. on Acoustics, Speech and Signal Processing **ASSP-27**, **2**, pp. 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, pp. 208–211.

CAPPE O. (1994) Elimination of Musical Noise Phenomenon with the Ephraim and Malah Noise Suppressor. IEEE Trans. Speech and Audio Processing, **2**, **2**, pp. 345–349.

CROZIER P.M. *et al* (1993) The Use of Linear Prediction and Spectral Scaling For Improving Speech Enhancement. EuroSpeech-93, pp. 231-234.

EPHRAIM Y. (1992) Statistical Model Based Speech Enhancement systems. Proc. IEEE, **80**, **10**, pp. 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, pp. 355–58.

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**, pp. 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.

KOBAYASHI T. *et al* (1993) Speech Recognition Under the Non-Stationary Noise Based on the Noise Hidden Markov Model and Spectral Subtraction. EuroSpeech-93, pp. 833–837.

LIM J.S. (1978) Evaluations of Correlation Subtraction Method for Enhancing Speech Degraded by Additive White Noise. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-26**, **5**, pp. 471–472.

LINHARD K. and KLEMM H. (1997) Noise Reduction with Spectral Subtraction and Median Filtering for Suppression of Musical Tones. Proc. ECSA-NATO Workshop on Robust Speech Recognition, pp. 159–162.

LOCKWOOD P. and BOUDY J. (1992) Experiments with a Non-linear Spectral Subtractor (NSS) Hidden Markov Models and the Projection, for Robust Speech Recognition in Car, Speech Communications. Elsevier, pp. 215–228.

LOCKWOOD P. *et al* (1992) Non-Linear Spectral Subtraction and Hidden Markov Models for Robust Speech Recognition in Car Noise Environments. ICASSP-92, pp. 265–268.

MILNER B.P. (1995) Speech Recognition in Adverse Environments. Ph.D. 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**, pp. 137–145, April.

NOLAZCO-FLORES J.A. and YOUNG S.J. (1994) Adapting a HMM-based Recogniser for Noisy Speech Enhanced by Spectral Subtraction. Proc. IEEE, Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP–94 Adelaide.

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, pp. 18A.2.1–18A.2.4.

O'SHAUGHNESSY D. (1989) Enhancing Speech Degraded by Additive Noise or Interfering Speakers. IEEE Commun. Mag. pp. 46–52.

POLLAK P. *et al* (1993) Noise Suppression System For A Car. EuroSpeech-93, pp. 1073–1076.

SORENSON H.B. (1993) Robust Speaker Independent Speech Recognition Using Non-Linear Spectral Subtraction Based IMELDA. EuroSpeech-93, pp. 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**, pp. 1847–1859.

VAN COMPERNOLLE D. (1989) Noise Adaptation in a Hidden Markov Model Speech Recognition System. Computer Speech and Language, **3**, pp. 151–167.

VASEGHI S.V. and FRAYLING-CORCK R. (1993) Restoration of Archived Gramophone Records, Journal of Audio Engineering Society.

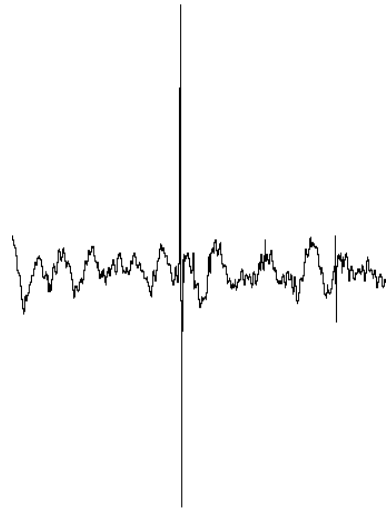
XIE F.(1993) Speech Enhancement by Non-Linear Spectral Estimation a Unifying Approach. EuroSpeech-93, pp. 617–620.

ZWICKER E. and FASTEL H. (1999) Psychoacoustics, Facts and Models, 2nd Ed. Springer.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

12



IMPULSIVE NOISE

Impulsive noise consists of relatively short duration "on/off" noise pulses, caused by a variety of sources, such as switching noise, adverse channel environments in a communication system, dropouts or surface degradation of audio recordings, clicks from computer keyboards, etc. An impulsive noise filter can be used for enhancing the quality and intelligibility of noisy signals, and for achieving robustness in pattern recognition and adaptive control systems. This chapter begins with a study of the frequency/time characteristics of impulsive noise, and then proceeds to consider several methods for statistical modelling of an impulsive noise process. The classical method for removal of impulsive noise is the median filter. However, the median filter often results in some signal degradation. For optimal performance, an impulsive noise removal system should utilise (a) the distinct features of the noise and the signal in the time and/or frequency domains, (b) the statistics of the signal and the noise processes, and (c) a model of the physiology of the signal and noise generation. We describe a model-based system that detects each impulsive noise, and then proceeds to replace the samples obliterated by an impulse. We also consider some methods for introducing robustness to impulsive noise in parameter estimation.

12.1 Impulsive Noise

In this section, first the mathematical concepts of an analog and a digital impulse are introduced, and then the various forms of real impulsive noise in communication systems are considered.

The mathematical concept of an analog impulse is illustrated in Figure 12.1. Consider the unit-area pulse $p(t)$ shown in Figure 12.1(a). As the pulse width Δ tends to zero, the pulse tends to an impulse. The impulse function shown in Figure 12.1(b) is defined as a pulse with an infinitesimal time width as

$$\delta(t) = \lim_{\Delta \to 0} p(t) = \begin{cases} 1/\Delta, & |t| \le \Delta/2 \\ 0, & |t| > \Delta/2 \end{cases} \tag{12.1}$$

The integral of the impulse function is given by

$$\int_{-\infty}^{\infty} \delta(t)\, dt = \Delta \times \frac{1}{\Delta} = 1 \tag{12.2}$$

The Fourier transform of the impulse function is obtained as

$$\Delta(f) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi ft} dt = e^0 = 1 \tag{12.3}$$

where f is the frequency variable. The impulse function is used as a *test function* to obtain the impulse response of a system. This is because as

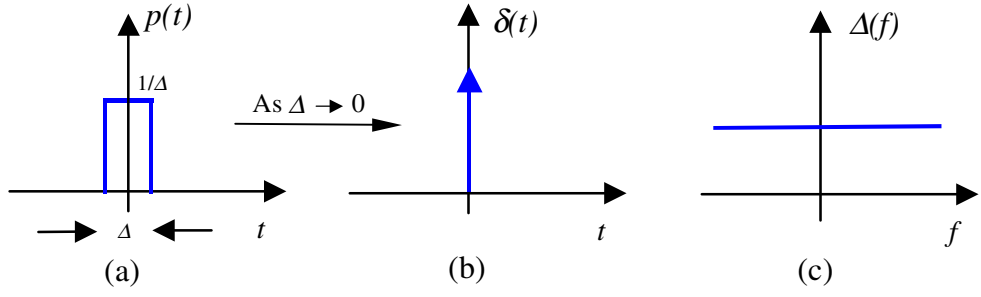


Figure 12.1 (a) A unit-area pulse, (b) The pulse becomes an impulse as $\Delta \to 0$, (c) The spectrum of the impulse function.

shown in Figure 12.1(c), *an impulse is a spectrally rich signal containing all frequencies in equal amounts*.

A digital impulse $\delta(m)$, shown Figure 12.2(a), is defined as a signal with an "on" duration of one sample, and is expressed as:

$$\delta(m)=\begin{cases}1, & m=0\\ 0, & m\neq 0\end{cases} \tag{12.4}$$

where the variable m designates the discrete-time index. Using the Fourier transform relation, the frequency spectrum of a digital impulse is given by

$$\Delta(f)=\sum_{m=-\infty}^{\infty}\delta(m)e^{-j2\pi fm}=1.0,\ -\infty<f<\infty \tag{12.5}$$

In communication systems, real impulsive-type noise has a duration that is normally more than one sample long. For example, in the context of audio signals, short-duration, sharp pulses, of up to 3 milliseconds (60 samples at a 20 kHz sampling rate) may be considered as impulsive-type noise. Figures 12.1(b) and 12.1(c) illustrate two examples of short-duration pulses and their respective spectra.

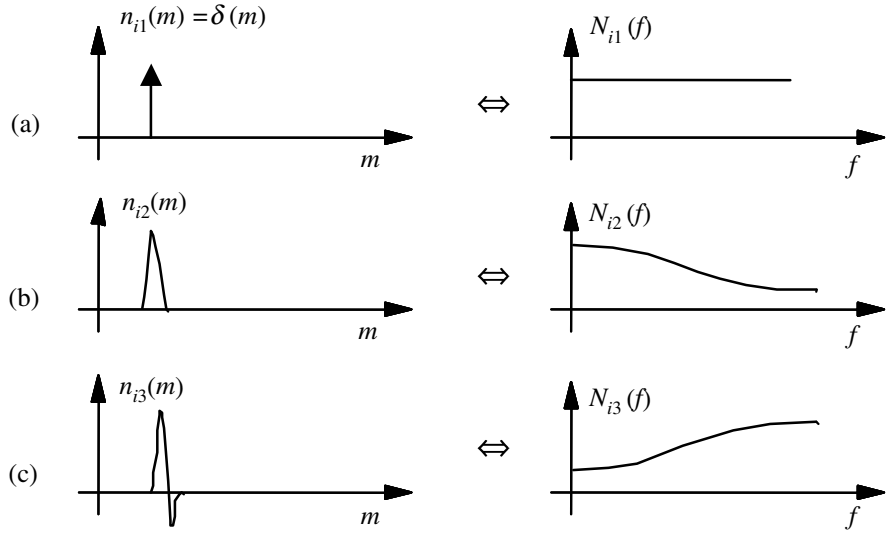


Figure 12.2 Time and frequency sketches of (a) an ideal impulse, and (b) and (c) short-duration pulses.

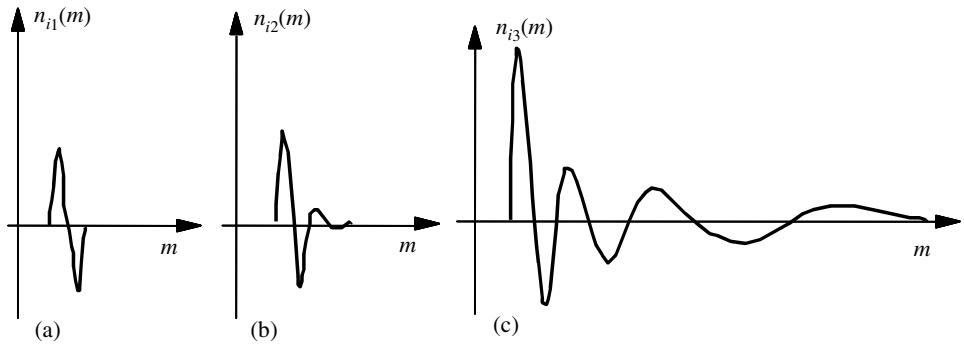


Figure 12.3 Illustration of variations of the impulse response of a non-linear system with increasing amplitude of the impulse.

In a communication system, an impulsive noise originates at some point in time and space, and then propagates through the channel to the receiver. The received noise is shaped by the channel, and can be considered as the channel impulse response. In general, the characteristics of a communication channel may be linear or non-linear, stationary or time varying. Furthermore, many communication systems, in response to a large-amplitude impulse, exhibit a nonlinear characteristic.

Figure 12.3 illustrates some examples of impulsive noise, typical of those observed on an old gramophone recording. In this case, the communication channel is the playback system, and may be assumed 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.

An important consideration in the development of a noise processing system is the choice of an appropriate domain (time or the frequency) for signal representation. The choice should depend on the specific objective of the system. In signal restoration, the objective is to separate the noise from the signal, and the representation domain must be the one that 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.

12.1.1 Autocorrelation and Power Spectrum of Impulsive Noise

Impulsive noise is a non-stationary, binary-state sequence of impulses with random amplitudes and random positions of occurrence. The non-stationary nature of impulsive noise can be seen by considering the power spectrum of a noise process with a few impulses per second: when the noise is absent the process has zero power, and when an impulse is present the noise power is the power of the impulse. Therefore the power spectrum and hence the autocorrelation of an impulsive noise is a binary state, time-varying process. An impulsive noise sequence can be modelled as an amplitude-modulated binary-state sequence, and expressed as

$$n_i(m) = n(m)\,b(m) \tag{12.6}$$

where $b(m)$ is a binary-state random sequence of ones and zeros, and $n(m)$ is a random noise process. Assuming that impulsive noise is an uncorrelated random process, the autocorrelation of impulsive noise may be defined as a binary-state process:

$$r_{nn}(k,m) = \mathcal{E}[n_i(m)n_i(m+k)] = \sigma_n^2\,\delta(k)b(m) \tag{12.7}$$

where $\delta(k)$ is the Kronecker delta function. Since it is assumed that the noise is an uncorrelated process, the autocorrelation is zero for $k \neq 0$, therefore Equation (12.7) may be written as

$$r_{nn}(0,m) = \sigma_n^2\,b(m) \tag{12.8}$$

Note that for a zero-mean noise process, $r_{nn}(0,m)$ is the time-varying binary-state noise power. The power spectrum of an impulsive noise sequence is obtained, by taking the Fourier transform of the autocorrelation function Equation (12.8), as

$$P_{N_I N_I}(f,m) = \sigma_n^2\,b(m) \tag{12.9}$$

In Equation (12.8) and (12.9) the autocorrelation and power spectrum are expressed as binary state functions that depend on the "on/off" state of impulsive noise at time m.

12.2 Statistical Models for Impulsive Noise

In this section, we study a number of statistical models for the characterisation of an impulsive noise process. An impulsive noise sequence $n_i(m)$ consists of short duration pulses of a random amplitude, duration, and time of occurrence, and may be modelled as the output of a filter excited by an amplitude-modulated random binary sequence as

$$n_i(m) = \sum_{k=0}^{P-1} h_k n(m-k) b(m-k) \tag{12.10}$$

Figure 12.4 illustrates the impulsive noise model of Equation (12.10). In Equation (12.10) $b(m)$ is a binary-valued random sequence model of the time of occurrence of impulsive noise, $n(m)$ is a continuous-valued random process model of impulse amplitude, and $h(m)$ is the impulse response of a filter that models the duration and shape of each impulse. Two important statistical processes for modelling impulsive noise as an amplitude-modulated binary sequence are the Bernoulli-Gaussian process and the Poisson–Gaussian process, which are discussed next.

12.2.1 Bernoulli–Gaussian Model of Impulsive Noise

In a Bernoulli-Gaussian model of an impulsive noise process, the random time of occurrence of the impulses is modelled by a binary Bernoulli process $b(m)$ and the amplitude of the impulses is modelled by a Gaussian

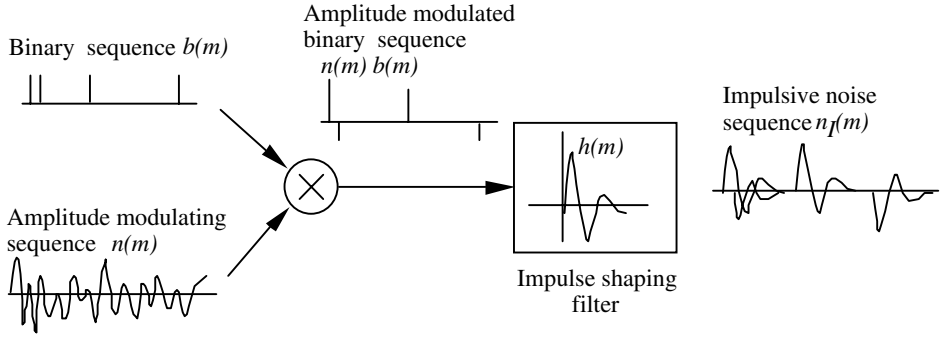


Figure 12.4 Illustration of an impulsive noise model as the output of a filter excited by an amplitude-modulated binary sequence.

process $n(m)$. A Bernoulli process $b(m)$ is a binary-valued process that takes a value of "1" with a probability of α and a value of "0" with a probability of $1-\alpha$. The probability mass function of a Bernoulli process is given by

$$P_B(b(m)) = \begin{cases} \alpha & \text{for } b(m)=1 \\ 1-\alpha & \text{for } b(m)=0. \end{cases} \tag{12.11}$$

A Bernoulli process has a mean

$$\mu_b = \mathcal{E}[(b(m))] = \alpha \tag{12.12}$$

and a variance

$$\sigma_b^2 = \mathcal{E}\left[(b(m)-\mu_b)^2\right] = \alpha(1-\alpha) \tag{12.13}$$

A zero-mean Gaussian pdf model of the random amplitudes of impulsive noise is given by

$$f_N(n(m)) = \frac{1}{\sqrt{2\pi}\,\sigma_n} \exp\left[-\frac{n^2(m)}{2\sigma_n^2}\right] \tag{12.14}$$

where σ_n^2 is the variance of the noise amplitude. In a Bernoulli–Gaussian model the probability density function of an impulsive noise $n_i(m)$ is given by

$$f_N^{BG}(n_i(m)) = (1-\alpha)\delta(n_i(m)) + \alpha f_N(n_i(m)) \tag{12.15}$$

where $\delta(n_i(m))$ is the Kronecker delta function. Note that the function $f_N^{BG}(n_i(m))$ is a mixture of a discrete probability mass function $\delta(n_i(m))$ and a continuous probability density function $f_N(n_i(m))$.

An alternative model for impulsive noise is a binary-state Gaussian process (Section 2.5.4), with a low-variance state modelling the absence of impulses and a relatively high-variance state modelling the amplitude of impulsive noise.

12.2.2 Poisson–Gaussian Model of Impulsive Noise

In a Poisson–Gaussian model the probability of occurrence of an impulsive noise event is modelled by a Poisson process, and the distribution of the random amplitude of impulsive noise is modelled by a Gaussian process. The Poisson process, described in Chapter 2, is a random event-counting process. In a Poisson model, the probability of occurrence of k impulsive noise in a time interval of T is given by

$$P(k,T)=\frac{(\lambda T)^k}{k!}e^{-\lambda T} \tag{12.16}$$

where λ is a rate function with the following properties:

$$\begin{aligned} &Prob(\text{one impulse in a small time interval } \Delta t)=\lambda\Delta t \\ &Prob(\text{zero impulse in a small time interval } \Delta t)=1-\lambda\Delta t \end{aligned} \tag{12.17}$$

It is assumed that no more than one impulsive noise can occur in a time interval Δt. In a Poisson–Gaussian model, the pdf of an impulsive noise $n_i(m)$ in a small time interval of Δt is given by

$$f_{N_I}^{PG}\left(n_i(m)\right)=(1-\lambda\Delta t)\,\delta\left(n_i(m)\right)+\lambda\Delta t\, f_N\left(n_i(m)\right) \tag{12.18}$$

where $f_N\left(n_i(m)\right)$ is the Gaussian pdf of Equation (12.14).

12.2.3 A Binary-State Model of Impulsive Noise

An impulsive noise process may be modelled by a binary-state model as shown in Figure 12.4. In this binary model, the state S_0 corresponds to the "off" condition when impulsive noise is absent; in this state, the model emits zero-valued samples. The state S_1 corresponds to the "on" condition; in this state the model emits short-duration pulses of random amplitude and duration. The probability of a transition from state S_i to state S_j is denoted by a_{ij}. In its simplest form, as shown in Figure 12.5, the model is memoryless, and the probability of a transition to state S_i is independent of the current state of the model. In this case, the probability that at time $t+1$

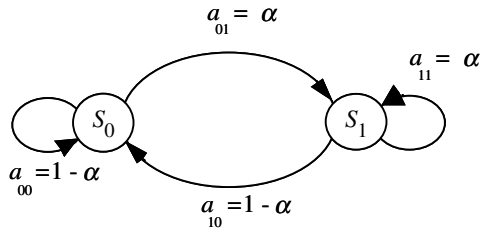


Figure 12.5 A binary-state model of an impulsive noise generator.

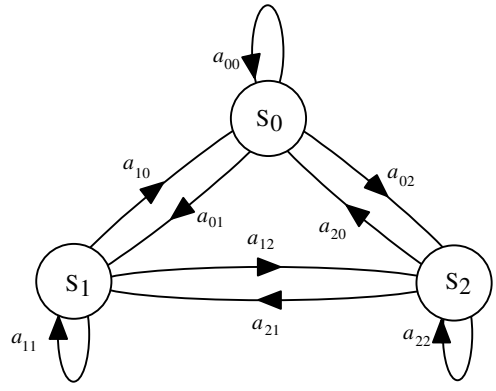


Figure 12.6 A 3-state model of impulsive noise and the decaying oscillations that often follow the impulses.

the signal is in the state S_0 is independent of the state at time t, and is given by

$$P\left(s(t+1)=S_0\middle|s(t)=S_0\right)=P\left(s(t+1)=S_0\middle|s(t)=S_1\right)=1-\alpha \qquad (12.19)$$

where s_t denotes the state at time t. Likewise, the probability that at time $t+1$ the model is in state S_1 is given by

$$P\left(s(t+1)=S_1\middle|s(t)=S_0\right)=P\left(s(t+1)=S_1\middle|s(t)=S_1\right)=\alpha \qquad (12.20)$$

In a more general form of the binary-state model, a Markovian state-transition can model the dependencies in the noise process. The model then becomes a 2-state hidden Markov model considered in Chapter 5.

In one of its simplest forms, the state S_1 emits samples from a zero-mean Gaussian random process. The impulsive noise model in state S_1 can be configured to accommodate a variety of impulsive noise of different shapes,

durations and pdfs. A practical method for modelling a variety of impulsive noise is to use a code book of M prototype impulsive noises, and their associated probabilities $[(n_{i1}, p_{i1}), (n_{i2}, p_{i2}), ..., (n_{iM}, p_{iM})]$, where p_j denotes the probability of impulsive noise of the type n_j. The impulsive noise code book may be designed by classification of a large number of "training" impulsive noises into a relatively small number of clusters. For each cluster, the average impulsive noise is chosen as the representative of the cluster. The number of impulses in the cluster of type j divided by the total number of impulses in all clusters gives p_j, the probability of an impulse of type j.

Figure 12.6 shows a three-state model of the impulsive noise and the decaying oscillations that might follow the noise. In this model, the state S_0 models the absence of impulsive noise, the state S_1 models the impulsive noise and the state S_2 models any oscillations that may follow a noise pulse.

12.2.4 Signal to Impulsive Noise Ratio

For impulsive noise the average signal to impulsive noise ratio, averaged over an entire noise sequence including the time instances when the impulses are absent, depends on two parameters: (a) the average power of each impulsive noise, and (b) the rate of occurrence of impulsive noise. Let P_{impulse} denote the average power of each impulse, and P_{signal} the signal power. We may define a "local" time-varying signal to impulsive noise ratio as

$$SINR(m) = \frac{P_{\text{signal}}(m)}{P_{\text{impulse}}\, b(m)} \tag{12.21}$$

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

$$SINR = \frac{P_{\text{signal}}}{\alpha\, P_{\text{impulse}}} \tag{12.22}$$

Note that from Equation (12.22), for a given signal power, there are many pair of values of α and P_{Impulse} that can yield the same average SINR.

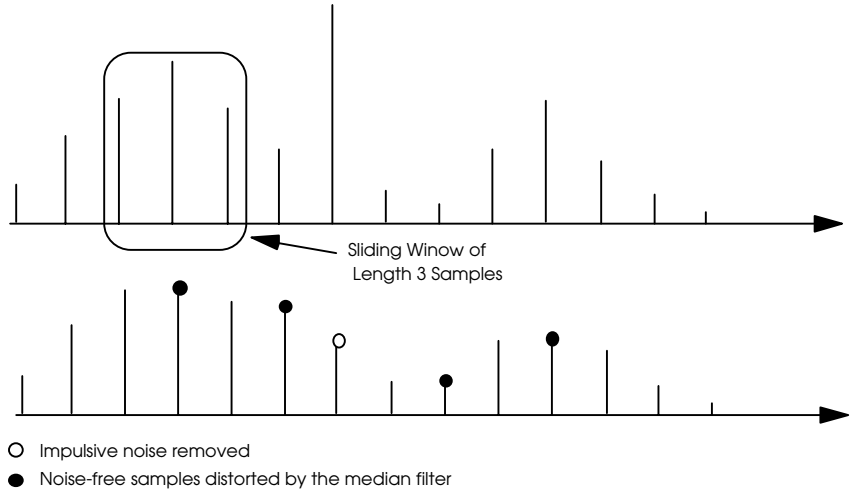


Figure 12.7 Input and output of a median filter. Note that in addition to suppressing the impulsive outlier, the filter also distorts some genuine signal components.

12.3 Median Filters

The classical approach to removal of impulsive noise is the median filter. The median of a set of samples $\{x(m)\}$ is a member of the set $x_{\text{med}}(m)$ such that; half the population of the set 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 12.7.

The output $\hat{x}(m)$ of a median filter with input $y(m)$ and a median window of length $2K+1$ samples is given by

$$\begin{aligned}\hat{x}(m) &= y_{\text{med}}(m) \\ &= \text{median}\big[y(m-K),\ldots,y(m),\ldots,y(m+K)\big]\end{aligned} \tag{12.23}$$

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 preserves edges or stepwise discontinuities in the signal. Median filters can be used for removing impulses in an image without smearing the edge information; this is of significant importance in image processing. However, experiments with median filters, for removal of impulsive noise from audio signals, demonstrate that median filters are unable to produce high-quality audio restoration. The median filters cannot deal with "real" impulsive noise, which 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:

$$\hat{x}(m)=\begin{cases} y(m) & \text{if } \left|y(m)-y_{\text{med}}(m)\right| < k\,\theta(m) \\ y_{\text{med}}(m) & \text{otherwise} \end{cases} \tag{12.24}$$

where $\theta(m)$ is an adaptive threshold that may be related to a robust estimate of the average of $\left|y(m)-y_{\text{med}}(m)\right|$, 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 a autoregressive model-based impulsive removal system, capable of producing high-quality audio restoration.

12.4 Impulsive Noise Removal Using Linear Prediction Models

In this section, we study a model-based impulsive noise removal system. Impulsive disturbances usually contaminate a relatively small fraction α of the total samples. Since a large fraction, $1-\alpha$, of samples 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 disturbed by impulsive noise. The impulsive noise removal system shown in Figure 12.8 consists of two subsystems: a detector and an interpolator. The detector locates the position of each noise pulse, and the interpolator replaces the distorted samples

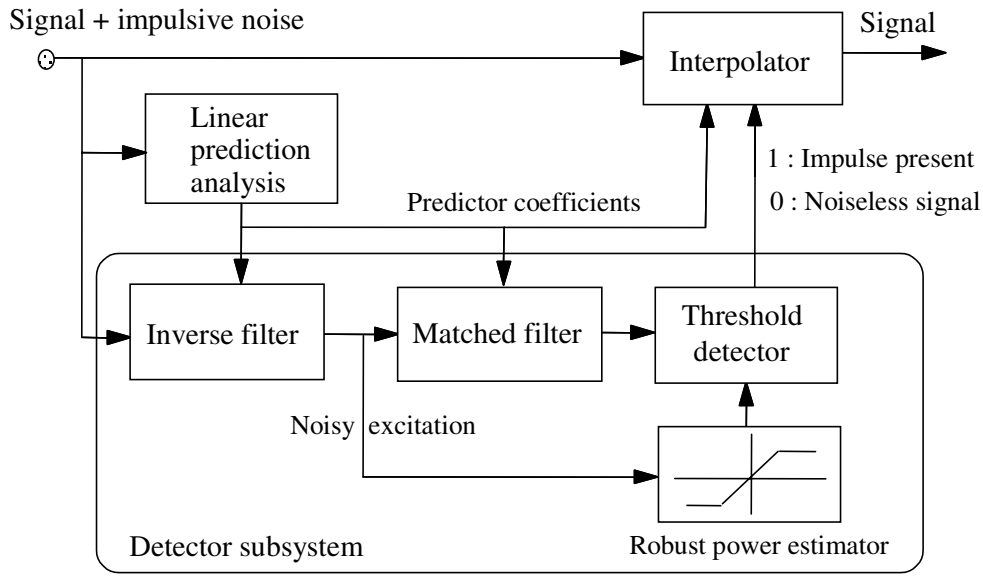


Figure 12.8 Configuration of an impulsive noise removal system incorporating a detector and interpolator subsystems.

using the samples on both sides of the impulsive noise. The detector is composed of a linear prediction analysis system, a matched filter and a threshold detector. The output of the detector is a binary switch and controls the interpolator. A detector output of "0" signals the absence of impulsive noise and the interpolator is bypassed. A detector output of "1" signals the presence of impulsive noise, and the interpolator is activated to replace the samples obliterated by noise.

12.4.1 Impulsive Noise Detection

A simple method for detection of impulsive noise is to employ an amplitude threshold, and classify those samples with an amplitude 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 12.9 shows a model for a noisy signal. The noise-free signal $x(m)$ is described by a linear prediction model as

$$x(m)=\sum_{k=1}^{P} a_k\, x(m-k)+e(m) \tag{12.25}$$

where $\boldsymbol{a}=[a_1,\ a_2,\ ...,a_P]^{\mathrm{T}}$ 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 12.9. The impulsive noise detector is based on the observation that linear predictors are a good model of the correlated signals but not the uncorrelated binary-state impulsive-type noise. Transforming the noisy signal $y(m)$ to the excitation signal of the predictor has the following effects:

(a) 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.
(b) The signal is decorrelated, whereas the impulsive noise is smeared and transformed to a scaled version of the impulse response of the inverse filter.

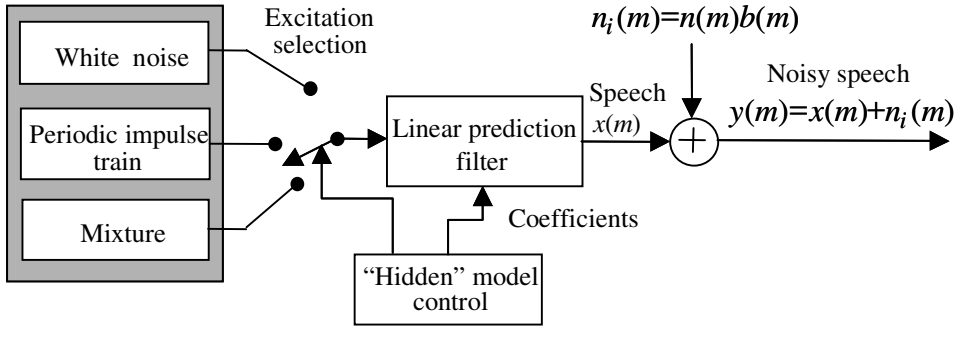


Figure 12.9 Noisy speech model. The signal is modelled by a linear predictor. Impulsive noise is modelled as an amplitude-modulated binary-state process.

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.

12.4.2 Analysis of Improvement in Noise Detectability

In the following, the improvement in noise detectability that results from inverse filtering is analysed. Using Equation (12.25), we can rewrite a noisy signal model as

$$\begin{aligned} y(m) &= x(m) + n_i(m) \\ &= \sum_{k=1}^{P} a_k\, x(m-k) + e(m) + n_i(m) \end{aligned} \tag{12.26}$$

where $y(m)$, $x(m)$ and $n_i(m)$ are the noisy signal, the signal and the noise respectively. Using an estimate $\hat{\boldsymbol{a}}$ 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

$$\begin{aligned} v(m) &= y(m) - \sum_{k=1}^{P} \hat{a}_k\, y(m-k) \\ &= x(m) + n_i(m) - \sum_{k=1}^{P} (a_k - \tilde{a}_k)[x(m-k) + n_i(m-k)] \end{aligned} \tag{12.27}$$

where $\tilde{a}_k$ is the error in the estimate of the predictor coefficient. Using Equation (12.25) Equation (12.27) can be rewritten in the following form:

$$v(m) = e(m) + n_i(m) + \sum_{k=1}^{P} \tilde{a}_k x(m-k) - \sum_{k=1}^{P} \hat{a}_k n_i(m-k) \tag{12.28}$$

From Equation (12.28) there are essentially three terms that contribute to the noise in the excitation sequence:

(a) the impulsive disturbance $n_i(m)$ which is usually the dominant term;
(b) the effect of the past P noise samples, smeared to the present time by the action of the inverse filtering, $\sum \hat{a}_k n_i(m-k)$;
(c) the increase in the variance of the excitation signal, caused by the error in the parameter vector estimate, and expressed by the term $\sum \tilde{a}_k x(m-k)$.

The improvement resulting from the inverse filter can be formulated as follows. The impulsive noise to signal ratio for the noisy signal is given by

$$\frac{\text{impulsive noise power}}{\text{signal power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[x^2(m)]} \tag{12.29}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. Note that in impulsive noise detection, the signal of interest is the impulsive noise to be detected from the accompanying signal. Assuming that the dominant noise term in the noisy excitation signal $v(m)$ is the impulse $n_i(m)$, the impulsive noise to excitation signal ratio is given by

$$\frac{\text{impulsive noise power}}{\text{excitation power}} = \frac{\mathcal{E}[n_i^2(m)]}{\mathcal{E}[e^2(m)]} \tag{12.30}$$

The overall gain in impulsive noise to signal ratio is obtained, by dividing Equations (12.29) and (12.30), as

$$\frac{\mathcal{E}[x^2(m)]}{\mathcal{E}[e^2(m)]} = \text{gain} \tag{12.31}$$

This simple analysis demonstrates that the improvement in impulsive noise detectability depends on the power amplification characteristics, due to resonances, of the linear predictor model. For speech signals, the scale of the amplitude of the noiseless speech excitation is on the order of 10^{-1} to 10^{-4} 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 12.10 illustrates the effect of inverse filtering in improving the detectability of impulsive noise. The inverse filtering has the effect that the signal $x(m)$ is transformed to an uncorrelated excitation signal $e(m)$, whereas the impulsive noise is smeared to a scaled version of the inverse filter impulse response $[1, -a_1, ..., -a_P]$, as indicated by the term $\sum \hat{a}_k n_i(m-k)$ in Equation (12.28). Assuming that the excitation is a white noise Gaussian signal, a filter matched to the inverse filter coefficients may enhance the delectability of the smeared impulsive noise from the excitation signal.

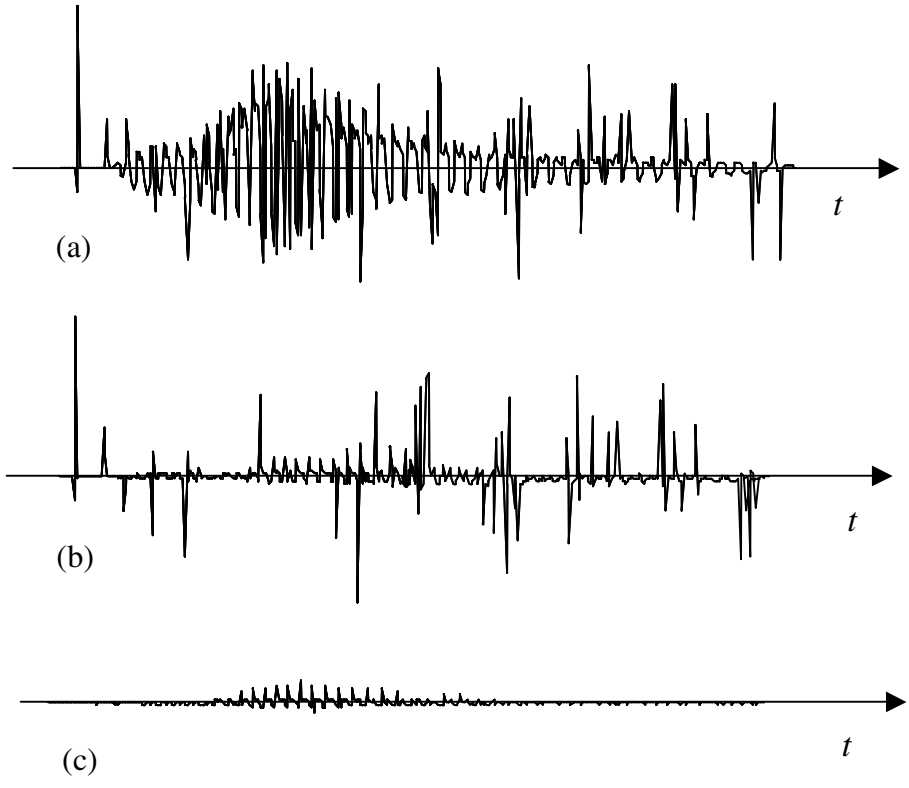


Figure 12.10 Illustration of the effects of inverse filtering on detectability of Impulsive noise: (a) Impulsive noise contaminated speech with 5% impulse contamination at an average SINR of 10dB, (b) Speech excitation of impulse-contaminated speech, and (c) Speech excitation of impulse-free speech.

12.4.3 Two-Sided Predictor for Impulsive Noise Detection

In the previous section, it was shown that impulsive noise detectability can be improved by decorrelating the speech signal. The process of decorrelation can be taken further by the use of a two-sided linear prediction model. The two-sided linear prediction of a sample $x(m)$ is based on the P past samples and the P future samples, and is defined by the equation

$$x(m)=\sum_{k=1}^{P} a_k x(m-k)+\sum_{k=1}^{P} a_{k+P} x(m+k)+e(m) \tag{12.32}$$

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 (12.32) the correlations of each sample with the future, as well as the past, samples are modeled. Although Equation (12.32) 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.

12.4.4 Interpolation of Discarded Samples

Samples irrevocably distorted by an impulsive noise are discarded and the gap thus left is interpolated. For interpolation imperfections to remain inaudible a high-fidelity interpolator is required. A number of interpolators for replacement of a sequence of missing samples are introduced in Chapter 10. The least square autoregressive (LSAR) interpolation algorithm of Section 10.3.2 produces high-quality results for a relatively small number of missing samples left by an impulsive noise. The LSAR interpolation method is a two-stage process. In the first stage, the available samples on both sides of the noise pulse are used to estimate the parameters of a linear prediction model of the signal. In the second stage, the estimated model parameters, and the samples on both sides of the gap are used to interpolate the missing samples. The use of this interpolator in replacement of audio signals distorted by impulsive noise has produced high-quality results.

12.5 Robust Parameter Estimation

In Figure 12.8, the threshold used for detection of impulsive noise from the excitation signal is derived from a nonlinear robust estimate of the excitation power. In this section, we consider 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

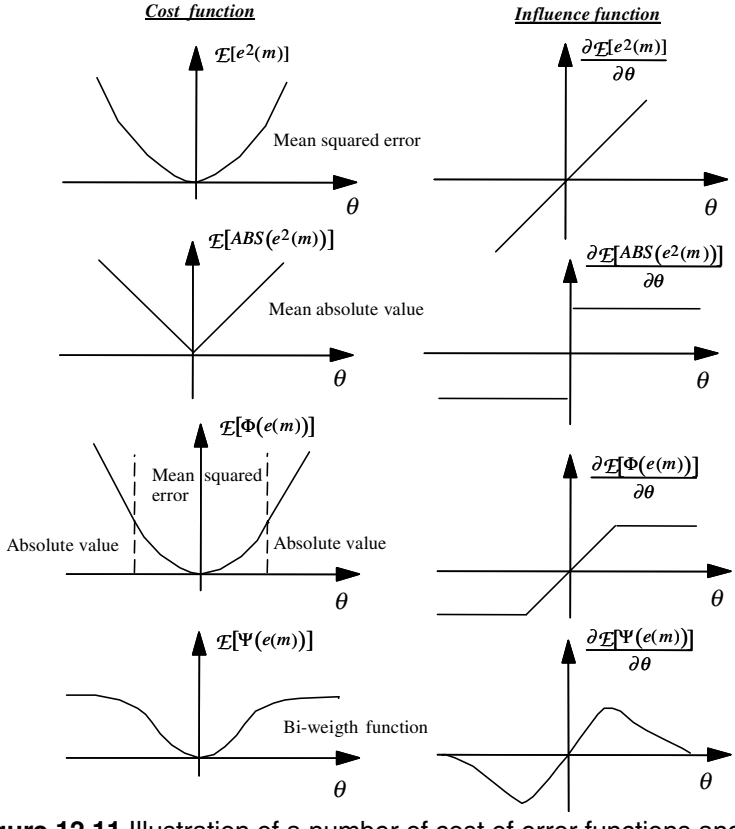


Figure 12.11 Illustration of a number of cost of error functions and the corresponding influence functions.

overshadow the influence of noise-free samples.

Figure 12.11 illustrates the variations of several cost of error functions with a parameter θ. Figure 12.11(a) shows a least square error cost function and its influence function. The influence function is the derivative of the cost function, and, as the name implies, it has a direct influence on the estimation results. It can be seen from the influence function of Figure 12.11(a) that an unbounded sample has an unbounded influence on the estimation results.

A method for introducing robustness is to use a 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 12.11(b). One disadvantage of this function is that it is not continuous at the origin. A further drawback is that it does not allow for the fact that, in practice, a large proportion of the samples are 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

$$\psi[e(m)] = \begin{cases} e^2(m) & \text{if } |e(m)| \le k \\ k|e(m)| & \text{otherwise} \end{cases} \tag{12.33}$$

Huber's function, shown in Figure 12.11(c), is a hybrid of the least mean square and the absolute value of error functions. Tukeys bi-weight function, which is a redescending robust objective function, is defined as

$$\psi[e(m)] = \begin{cases} \{1-[1-e^2(m)]^3\}/6 & \text{if } |e(m)| \le 1 \\ 1/6 & \text{otherwise} \end{cases} \tag{12.34}$$

As shown in Figure 12.11(d), the influence function is linear for small signal values but introduces attenuation as the signal value exceeds some threshold. The threshold may be obtained from a robust median estimate of the signal power.

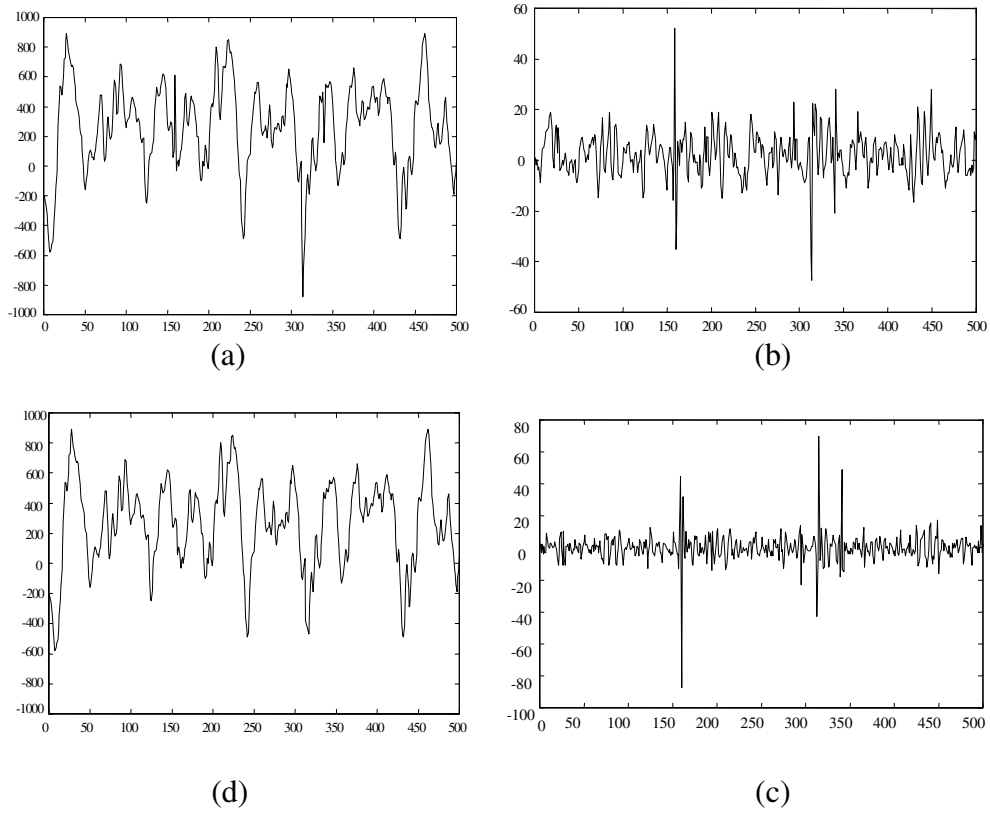


Figure 12.12 (a) A noisy audio signal from a 78 rpm record, (b) Noisy excitation signal, (c) Matched filter output, (d) Restored signal.

12.6 Restoration of Archived Gramophone Records

This Section describes the application of the impulsive noise removal system of Figure 12.8 to the restoration of archived audio records. As the bandwidth of archived recordings is limited to 7–8 kHz, a low-pass, anti-aliasing filter with a cutoff frequency of 8 kHz is used to remove the out of band noise. Playedback signals were sampled at a rate of 20 kHz, and digitised to 16 bits. Figure 12.12(a) shows a 25 ms segment of noisy music and song from an old 78 rpm gramophone record. The impulsive interferences are due to faults in the record stamping process, granularities of the record material or physical damage. This signal is modelled by a predictor of order 20. The excitation signal obtained from the inverse filter and the matched filter output are shown in Figures 12.12(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.

The amplitude threshold for detection of impulsive noise from the excitation signal is adapted on a block basis, and is set to $k\sigma_e^2$, where σ_e^2 is a robust estimate of the excitation power. The robust estimate is obtained by passing the noisy excitation signal through a soft nonlinearity that rejects outliers. The scalar k is a tuning parameter; the choice of k reflects a trade-off between the hit rate and the false-alarm rate of the detector. As k decreases, smaller noise pulses are detected but the false detection rate also increases. When an impulse is detected, a few samples are discarded and replaced by the LSAR interpolation algorithm described in Chapter 10. Figure 12.12(d) shows the signal with the impulses removed. The impulsive noise removal system of Figure 12.8 was successfully applied to restoration of numerous examples of archived gramophone records. The system is also effective in suppressing impulsive noise in examples of noisy telephone conversations.

12.7 Summary

The classic linear time-invariant theory on which many signal processing methods are based is not suitable for dealing with the 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 12.4, we considered an algorithm based on a linear prediction model of the signal for detection of impulsive noise.

In the next Chapter we expand the materials we considered 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**, pp. 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**, pp. 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. pp. 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**, pp. 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**, pp. 600–609, June.

MILNER B.P. (1995) Speech Recognition in Adverse Environments. PhD Thesis, University of East Anglia, UK.

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, pp. 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.

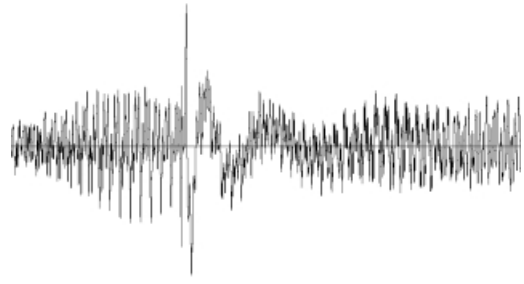
VASEGHI S.V. and RAYNER P.J.W. (1990) Detection and Suppression of Impulsive Noise in Speech Communication Systems. IEE Proc-I Communications Speech and Vision, pp. 38–46, February.

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, pp. 437–440.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

13



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 12 on impulsive noise.

13.1 Transient Noise Waveforms

Transient noise pulses often consist of a relatively short sharp initial pulse followed by decaying low-frequency oscillations as shown in Figure 13.1. The initial pulse is usually 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 characterize the transient noise pulses with a similar degree of consistency to that of characterizing the channels through which the pulses propagate.

As an illustration of the distribution of a transient noise pulse in time and frequency, consider the scratch pulses from a damaged gramophone record shown in Figures 13.1 and 13.2. Scratch noise pulses are acoustic manifestations of the response of the stylus and the associated 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 13.1, a typical scratch waveform often exhibits two distinct regions:

(a) the initial high-amplitude pulse response of the playback system to the physical discontinuity on the record medium; this is followed by
(b) decaying oscillations that cause additive distortion.

The initial pulse is relatively short and has a duration on the order of 1–5 ms, whereas the oscillatory tail has a longer duration and may last up to 50 ms. Note in Figure 13.1 that the frequency of the decaying oscillations decreases with time. This behaviour may be attributed to the nonlinear modes of response of the 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.

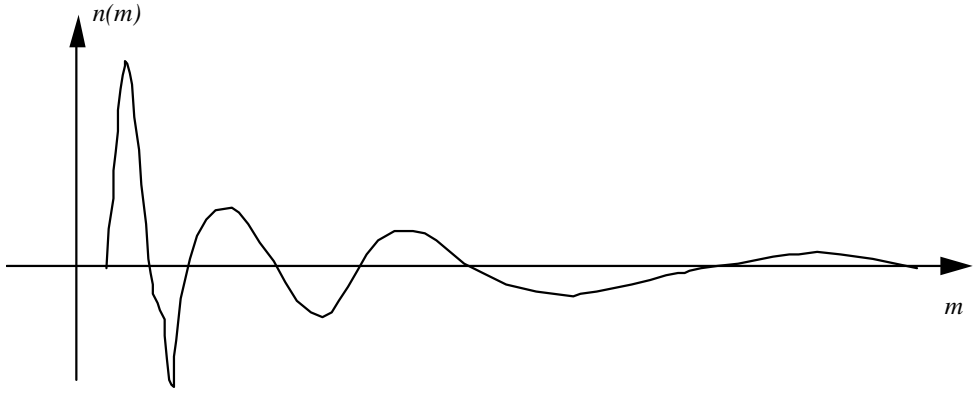
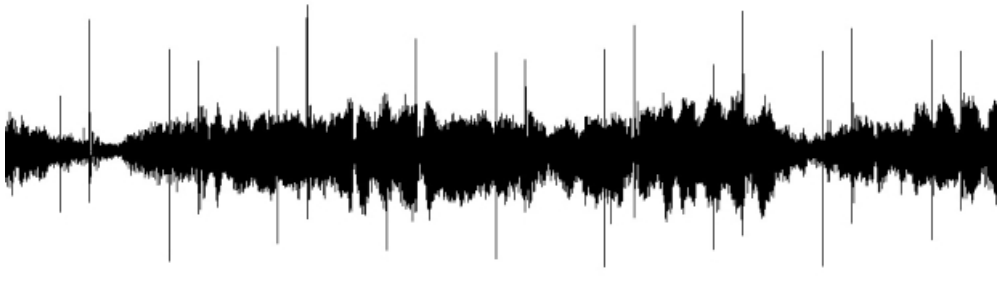
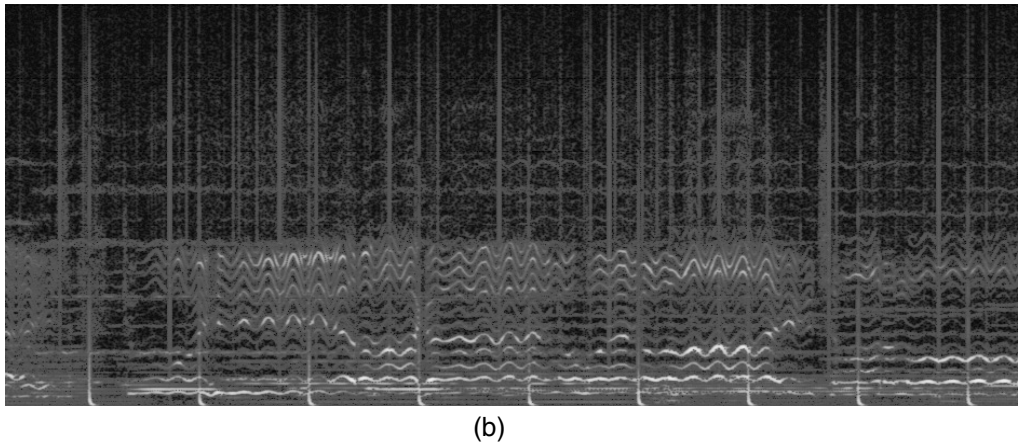


Figure 13.1 The profile of a transient noise pulse from a scratched gramophone record.



(a)



(b)

Figure 13.2 An example of (a) the time-domain waveform and (b) the spectrogram of transient noise scratch pulses in a damaged gramophone record.

A similar argument can be used to describe the transient noise pulses in other systems as the response of the system to an impulsive noise. Figure 13.2(a) (b) show the time-domain waveform and the spectrogram of a section of music and song with scratch-type noise. Note that as the scratch defect on the record was radial, the scratch pulses occure 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.

13.2 Transient Noise Pulse Models

To a first approximation, a transient noise pulse $n(m)$ can be modelled as the impulse response of a linear time-invariant filter model of the channel as

$$n(m)=\sum_k h_k A\delta(m-k)=Ah_m \tag{13.1}$$

where A is the amplitude of the driving impulse and h_k is the channel impulse response. A burst of overlapping, or closely spaced, noise pulses can be modelled as the response of a channel to a sequence of impulses as

$$n(m)=\sum_k h_k \sum_j A_j \delta\big((m-T_j)-k\big)=\sum_j A_j h_{m-T_j} \tag{13.2}$$

where it is assumed that the j^{th} transient pulse is due to an impulse of amplitude A_j at time T_j. In practice, a noise model should be able to deal with the statistical variations of a variety of noise and channel types. In this section, we consider three methods for modelling the temporal, spectral and durational characteristics of a transient noise pulse process:

(a) a template-based model;
(b) a linear-predictive model;
(c) a hidden Markov model.

13.2.1 Noise Pulse Templates

A widely used method for modelling the space of a random process is to model the process as a collection of signal clusters, and to design a 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 13.3 shows a number of typical noise pulses. The design of a code book of signal templates is described in Chapter 4.

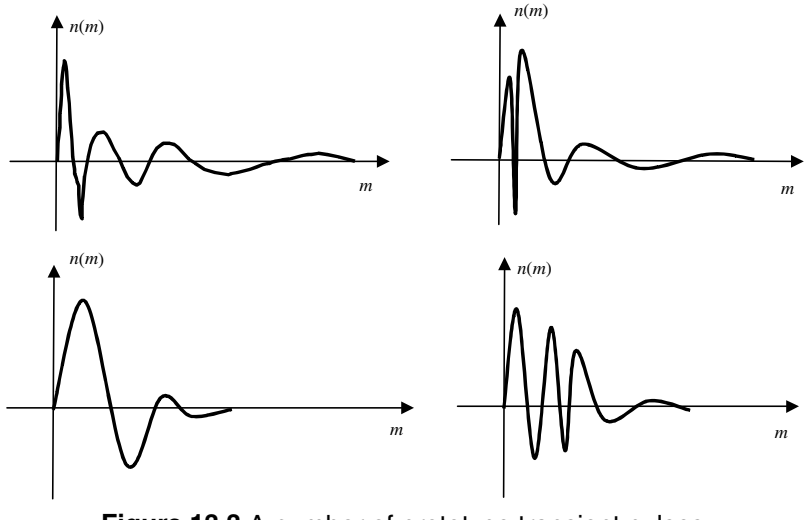


Figure 13.3 A number of prototype transient pulses.

13.2.2 Autoregressive Model of Transient Noise Pulses

Model-based methods have the advantage over template-based methods that overlapped noise pulses can be modelled as the response of the model to a number of closely spaced impulsive inputs. In this section, we consider an autoregressive (AR) model of transient noise pulses. The AR model for a single noise pulse $n(m)$ can be described as

$$n(m)=\sum_{k=1}^{P} c_k n(m-k)+A\delta(m) \tag{13.3}$$

where c_k are the AR model coefficients, 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:

$$n(m)=\sum_{k=1}^{P} c_k n(m-k)+\sum_{j}^{M} A_j \delta(m-T_j) \tag{13.4}$$

where it is assumed that T_j is the start of the j^{th} pulse in a burst of M excitation pulses.

An improved AR model for transient noise, proposed by Godsill, is driven by a two-state excitation: in the state S_0, the excitation is a zero-mean Gaussian process of small variance σ_0^2, and in the state S_1, the excitation is a zero-mean Gaussian process of relatively larger variance $\sigma_1^2 >> \sigma_0^2$. 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

$$e_n(m)=\left[\sigma_1 b(m)+\sigma_0 \bar{b}(m)\right]u(m) \tag{13.5}$$

where $u(m)$ is an uncorrelated zero-mean unit-variance Gaussian process, and $b(m)$ indicates the state of the excitation signal: $b(m)=1$ indicates that the excitation has a variance of σ_1^2, and $b(m)=0$ (or its binary complement

$\bar{b}(m)=1$) indicates the excitation has a smaller variance of σ_0^2. The time-varying variance of $e_n(m)$ can be expressed as

$$\sigma_{e_n}^2(m)=\sigma_1^2 b(m)+\sigma_0^2\bar{b}(m) \tag{13.6}$$

Assuming that the excitation pattern $b(m)$ is given, and that the excitation amplitude is Gaussian, the pdf of an N-sample long noise pulse $\boldsymbol{n}$ is given by

$$f_N(\boldsymbol{n})=\frac{1}{(2\pi)^{N/2}\left|\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}\right|^{1/2}}\exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}^{-1}\boldsymbol{C}\boldsymbol{n}\right) \tag{13.7}$$

where $\boldsymbol{C}$ is a matrix of coefficients of the AR model of the noise (as described in Section 8.4), and $\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}$ is the diagonal covariance matrix of the input to the noise model. The diagonal elements of $\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}$ are given by Equation (13.6).

13.2.3 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:

(a) the periods during which there are no noise pulses;
(b) the initial, and often short and sharp, pulse of a transient noise;
(c) the decaying oscillatory tail of a transient pulse.

Figure 13.4 illustrates a three-state HMM of transient noise pulses. The state S_0 models the periods when the noise pulses are absent. In this state, the noise process may be zero-valued. This state can also be used to model a different noise process such as a white noise process. The state S_1 models the relatively sharp pulse that forms the initial part of many transient noise pulses. The state S_2 models the decaying oscillatory part of a noise pulse that usually follows the initial pulse of a transient noise. A code book of waveforms in states S_1 and S_2 can model a variety of different noise pulses. Note that in the HMM model of Figure 13.4, the self-loop transition

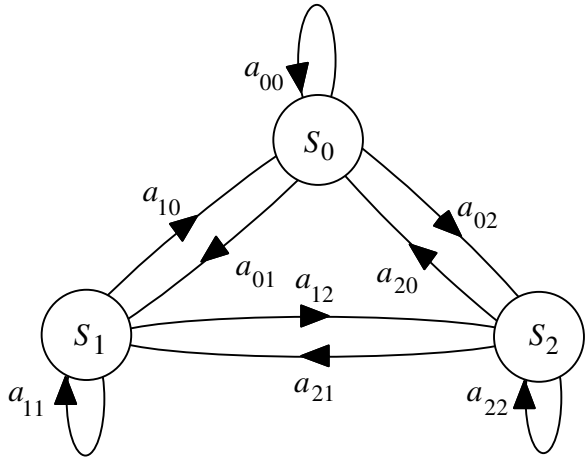


Figure 13.4 A three-state model of a transient noise pulse process.

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.

A hidden Markov model of noise can be employed for both the detection and the removal of transient noise pulses. As described in Section 13.3.3, the maximum-likelihood state-sequence of the noise HMM provides an estimate of the state of the noise at each time instant. The estimates of the states of the signal and the noise can be used for the implementation of an optimal state-dependent signal restoration algorithm.

13.3 Detection of Noise Pulses

For the detection of a pulse process $n(m)$ observed in an additive signal $x(m)$, the signal and the pulse can be modelled as

$$y(m)=b(m)n(m)+x(m) \tag{13.8}$$

where $b(m)$ is a binary "indicator" process that signals the presence or absence of a noise pulse. Using the model of Equation (13.8), the detection of a noise pulse process can be considered as the estimation of the underlying binary-state noise-indicator process $b(m)$. In this section, we

consider three different methods for detection of transient noise pulses, using the noise template model within a matched filter, the linear predictive model of noise, and the hidden Markov model described in Section 13.2.

13.3.1 Matched Filter for Noise Pulse Detection

The inner product of two signal vectors provides a measure of the similarity of the signals. Since filtering is basically an inner product operation, it follows that the output of a filter should provide a measure of similarity of the filter input and the filter impulse response. The classical method for detection of a signal is to use a filter whose impulse response is *matched* to the shape of the signal to be detected. The derivation of a matched filter for the detection of a pulse $n(m)$ is based on maximisation of the amplitude of the filter output when the input contains the pulse $n(m)$. The matched filter for the detection of a pulse $n(m)$ observed in a "background" signal $x(m)$ is defined as

$$H(f)=K\frac{N^*(f)}{P_{XX}(f)} \tag{13.9}$$

where $P_{XX}(f)$ is the power spectrum of $x(m)$ and $N^*(f)$ is the complex conjugate of the spectrum of the noise pulse. When the "background" signal process $x(m)$ is a zero mean uncorrelated signal with variance σ_x^2, the matched filter for detection of the transient noise pulse $n(m)$ becomes

$$H(f)=\frac{K}{\sigma_x^2}N^*(f) \tag{13.10}$$

The impulse response of the matched filter corresponding to Equation (13.10) is given by

$$h(m)=Cn(-m) \tag{13.11}$$

where the scaling factor C is given by $C = K/\sigma_x^2$. Let $z(m)$ denote the output of the matched filter. In response to an input noise pulse, the filter output is given by the convolution relation

$$z(m)=Cn(-m)*n(m) \tag{13.12}$$

where the asterisk * denotes convolution. In the frequency domain Equation (13.12) becomes

$$Z(f) = N(f)H(f) = C\left|N(f)\right|^2 \tag{13.13}$$

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

$$\hat{b}(m) = \begin{cases} 1 & \text{if } \left|z(m)\right| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{13.14}$$

In Equation (13.14), when the matched filter output exceeds a threshold, the detector flags the presence of the signal at the input. Figure 13.5 shows a noise pulse detector composed of a bank of M different matched filters. The detector signals the presence or the absence of a noise pulse. If a pulse is present then additional information 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. This information can be used for subtraction of the noise pulse from the noisy signal as described in Section 13.4.1.

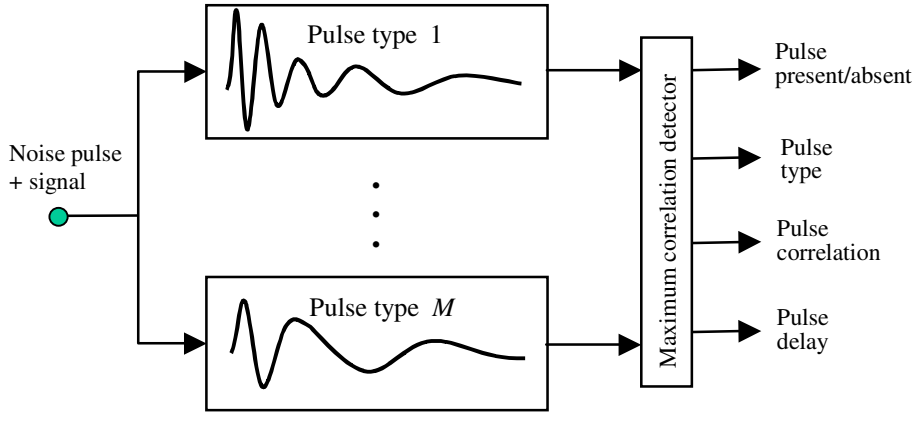


Figure 13.5 A bank of matched filters for detection of transient noise pulses.

13.3.2 Noise Detection Based on Inverse Filtering

The initial part of a transient noise pulse is often a relatively short and sharp impulsive-type event, which can be used as a distinctive feature for the detection of the noise pulses. The 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.

13.3.3 Noise Detection Based on HMM

In the three-state hidden Markov model of a transient noise pulse process, described in Section 13.2.3, the states S_0, S_1 and S_2 correspond to the noise-absent state, the initial noise pulse state, and the decaying oscillatory noise state respectively. As described in Chapter 5, an HMM, denoted by $\mathcal{M}$, is defined by a set of Markovian state transition probabilities and Gaussian state observation pdfs. The statistical parameters of the HMM of a noise pulse process can be obtained from a sufficiently large number of training examples of the process.

Given an observation vector $\boldsymbol{y}=[y(0), y(1), ..., y(N-1)]$, the maximum likelihood state sequence $\boldsymbol{s}=[s(0), s(1), ..., s(N-1)]$, of the HMM $\mathcal{M}$ is obtained as

$$\boldsymbol{s}_{ML} = \arg\max_{\boldsymbol{s}} f_{\boldsymbol{Y}|\boldsymbol{S}}(\boldsymbol{y} \mid \boldsymbol{s}, \mathcal{M}) \tag{13.15}$$

where, for a hidden Markov model, the likelihood of an observation sequence $f_{\boldsymbol{Y}|\boldsymbol{S}}(\boldsymbol{y}|\boldsymbol{s},\lambda)$ can be expressed as

$$f_{\boldsymbol{Y}|\boldsymbol{S}}\left(y(0),y(1),\ldots,y(N-1)\middle|s(0),s(1),\ldots,s(N-1)\right)$$
$$=\pi_{s(0)}f_{s(0)}\left(y(0)\right)a_{s(0),s(1)}f_{s(1)}\left(y(1)\right)\cdots a_{s(N-2),s(N-1)}f_{s(N-1)}\left(y(N-1)\right) \quad (13.16)$$

where $\pi_{s(i)}$ is the initial state probability, $a_{s(i),s(j)}$ is the probability of a transition from state $s(i)$ to state $s(j)$, and $f_{s(i)}(y(i))$ is the state observation pdf for the state $s(i)$. The maximum-likelihood state sequence $\boldsymbol{s}_{\mathrm{ML}}$, derived using the Viterbi algorithm, is an estimate of the underlying states of the noise pulse process, and can be used as a detector of the presence or absence of a noise pulse.

13.4 Removal of Noise Pulse Distortions

In this section, we consider two methods for the removal of transient noise pulses: (a) an adaptive noise subtraction method and (b) an autoregressive (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

13.4.1 Adaptive Subtraction of Noise Pulses

The transient noise removal system shown in Figure 13.6 is composed of a matched filter for detection of noise pulses, a linear adaptive noise subtractor for cancellation of the linear transitory part of a noise pulse, and an interpolator for the replacement of samples irrevocably distorted by the initial part of each pulse. Let $x(m)$, $n(m)$ and $y(m)$ denote the signal, the noise pulse and the noisy signal respectively; the noisy signal model is

$$y(m)=x(m)+b(m)\,n(m) \quad (13.17)$$

where the binary indicator sequence $b(m)$ indicates the presence or the absence of a noise pulse. Assume that each noise pulse $n(m)$ can be modelled as the amplitude-scaled and time-shifted version of the noise pulse template $\overline{n}(m)$ so that

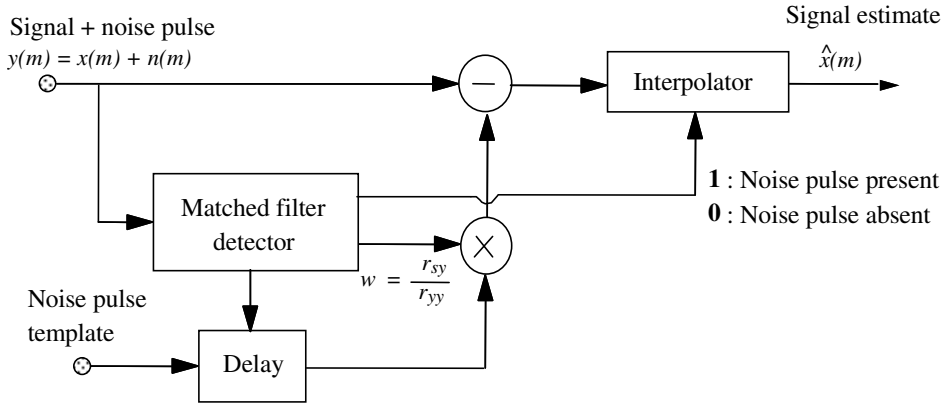


Figure 13.6 Transient noise pulse removal system.

$$n(m) \approx w\bar{n}(m-D) \tag{13.18}$$

where w is an amplitude scalar and the integer D denotes the relative delay (time shift) between the noise pulse template and the detected noise. From Equations (13.17) and (13.18) the noisy signal can be modelled:

$$y(m) \approx x(m) + w\bar{n}(m-D) \tag{13.19}$$

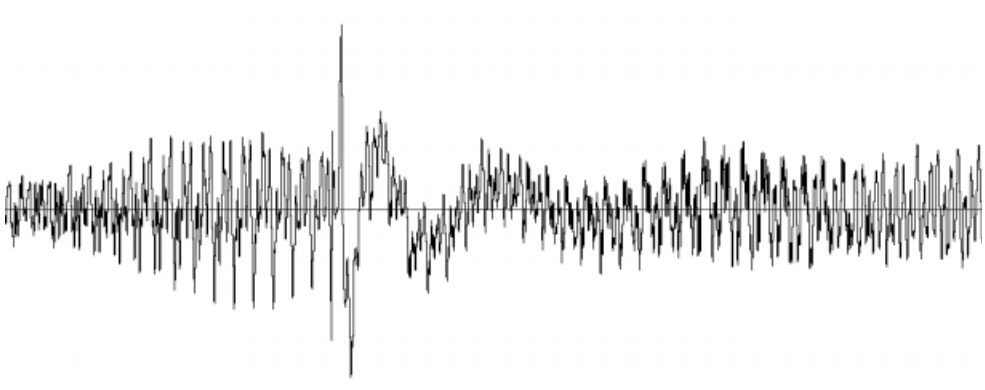
From Equation (13.19) an estimate of the signal $x(m)$ can be obtained by subtracting an estimate of the noise pulse from that of the noisy signal:

$$\hat{x}(m) = y(m) - w\bar{n}(m-D) \tag{13.20}$$

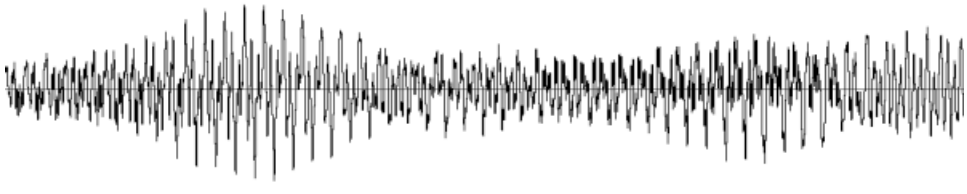
where the time delay D required for time-alignment of the noisy signal $y(m)$ and the noise template $\bar{n}(m)$ is obtained from the cross-correlation function CCF as

$$D = \arg\max_{k} \left[CCF\left(y(m), \bar{n}(m-k)\right) \right] \tag{13.21}$$

When a noise pulse is detected, the time lag corresponding to the maximum of the cross-correlation function is used to delay and time-align the noise pulse template with the noise pulse. The template energy is adaptively matched to that of the noise pulse by an adaptive scaling coefficient w. The scaled and time-aligned noise template is subtracted



(a)



(b)

Figure 13.7 (a) A signal from an old gramophone record with a scratch noise pulse. (b) The restored signal.

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 $\overline{n}(m-D)$ gives

$$\begin{aligned}\sum_{m=0}^{N-1} y(m)\overline{n}(m-D) &= \sum_{m=0}^{N-1}[x(m)+w\overline{n}(m-D)]\overline{n}(m-D) \\ &= \sum_{m=0}^{N-1} x(m)\overline{n}(m-D)+w\sum_{m=0}^{N-1}\overline{n}(m-D)\overline{n}(m-D)\end{aligned} \tag{13.22}$$

where N is the pulse template length. Since the signal $x(m)$ and the noise $n(m)$ are uncorrelated, the term $\Sigma\, x(m)\,\overline{n}(m-D)$ on the right hand side of Equation (13.22) is small, and we have

$$w \approx \frac{\sum_m x(m)\overline{n}(m-D)}{\sum_m \overline{n}^2(m-D)} \tag{13.23}$$

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 10. When there is no noise pulse, the coefficient w is zero, the interpolator is bypassed and the input signal is passed through unmodified. Figure 13.7(b) shows the result of processing the noisy signal of Figure 13.7(a). The linear oscillatory noise is completely removed by the adaptive subtraction method. For this signal 80 samples irrevocably distorted by the initial scratch pulse were discarded and interpolated.

13.4.2 AR-based Restoration of Signals Distorted by Noise Pulses

A model-based approach to noise detection/removal provides a more compact method for characterisation of transient noise pulses, and has the advantage that closely spaced pulses can be modelled as the response of the model to a number of closely spaced input impulses. The signal $x(m)$ is modelled as the output of an AR model of order P_1 as

$$x(m)=\sum_{k=1}^{P_1} a_k x(m-k)+e(m) \tag{13.24}$$

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

$$f_{\boldsymbol{X}}(\boldsymbol{x})=\frac{1}{(2\pi\sigma_e^2)^{N/2}}\exp\left(-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \tag{13.25}$$

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 (13.25), it is assumed that the P_1 initial samples are known. The AR model for a single noise pulse waveform $n(m)$ can be written as

$$n(m)=\sum_{k=1}^{P_2} c_k n(m-k)+A\delta(m) \tag{13.26}$$

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

$$n(m)=\sum_{k=1}^{P_2} a_k n(m-k)+\sum_{j}^{M} A_j \delta(m-T_j) \tag{13.27}$$

where it is assumed that T_k is the start of the k^{th} excitation pulse in a burst of M pulses. A linear predictor model proposed by Godsill is driven by a binary-state excitation. The excitation waveform has two states: in state "0", the excitation is a zero-mean Gaussian process of variance σ_0^2, and in state "1", the excitation is a zero-mean Gaussian process of variance $\sigma_1^2 >> \sigma_0^2$. 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

$$e_n(m)=\left[b(m)\sigma_1+\bar{b}(m)\sigma_0\right]u(m) \tag{13.28}$$

where $u(m)$ is an uncorrelated zero-mean Gaussian process of unit variance, $b(m)$ is a binary sequence that indicates the state of the excitation, and $\bar{b}(m)$ is the binary complement of $b(m)$. When $b(m)=1$ the excitation variance is σ_1^2 and when $b(m)=0$, the excitation variance is σ_0^2. The binary-state variance of $e_n(m)$ can be expressed as

$$\sigma_{e_n}^2(m)=b(m)\sigma_1^2+\bar{b}(m)\sigma_0^2 \tag{13.29}$$

Assuming that the excitation pattern $\boldsymbol{b}=[b(m)]$ is given, the pdf of an N sample noise pulse $\boldsymbol{x}$ is

$$f_N(\boldsymbol{n}\,|\,\boldsymbol{b})=\frac{1}{(2\pi)^{N/2}\left|\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}\right|^{1/2}}\exp\left(-\frac{1}{2}\boldsymbol{n}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}^{-1}\boldsymbol{C}\boldsymbol{n}\right) \tag{13.30}$$

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 $\boldsymbol{y}$, $f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y})$,can be expressed, using Bayes' rule, as

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}\,|\,\boldsymbol{y}) &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}\,|\,\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} f_{\boldsymbol{N}}(\boldsymbol{y}-\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) \end{aligned} \tag{13.31}$$

For a given observation $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is a constant. Substitution of Equations (13.30) and (13.25) in Equation (13.31) yields

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = &\frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})}\frac{1}{(2\pi\sigma_e)^N\left|\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}\right|^{1/2}} \\ &\times\exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}^{-1}\boldsymbol{C}(\boldsymbol{y}-\boldsymbol{x})-\frac{1}{2\sigma_e^2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\right) \end{aligned} \tag{13.32}$$

The MAP solution obtained by maximisation of the log posterior function with respect to the undistorted signal $\boldsymbol{x}$ is given by

$$\hat{\boldsymbol{x}}^{MAP}=\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}/\sigma_e^2+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{\Lambda}_{\boldsymbol{e}_n\boldsymbol{e}_n}^{-1}\boldsymbol{C}\,\boldsymbol{y} \tag{13.33}$$

13.5 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 13.2, we considered several transient noise pulse models including a template-based method, an AR model-based method and a hidden Markov model. In Sections 13.2 and 13.3, these models were applied to the detection and removal of noise pulses.

Bibliography

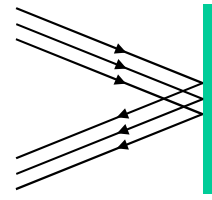
GODSILL S.J. (1993), The Restoration of Degraded Audio Signals. PhD Thesis, Cambridge University.

VASEGHI S.V. (1987), Algorithm for Restoration of Archived Gramophone Recordings. Ph.D. Thesis, Cambridge University.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

14



ECHO CANCELLATION

Echo is the repetition of a waveform due to reflection from points where the characteristics of the medium through which the wave propagates changes. Echo is usefully employed in sonar and radar for detection and exploration purposes. In telecommunication, echo can degrade the quality of service, and echo cancellation is an important part of communication systems. The development of echo reduction began in the late 1950s, and continues today as new integrated landline and wireless cellular networks put additional requirement on the performance of echo cancellers. There are two types of echo in communication systems: acoustic echo and telephone line hybrid echo. Acoustic echo results from a feedback path set up between the speaker and the microphone in a mobile phone, hands-free phone, teleconference or hearing aid system. Acoustic echo may be reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through different paths. Telephone line echoes result from an impedance mismatch at telephone exchange hybrids where the subscriber's 2-wire line is connected to a 4-wire line. The perceptual effects of an echo depend on the time delay between the incident and reflected waves, the strength of the reflected waves, and the number of paths through which the waves are reflected. Telephone line echoes, and acoustic feedback echoes in teleconference and hearing aid systems, are undesirable and annoying and can be disruptive. In this chapter we study some methods for removing line echo from telephone and data telecommunication systems, and acoustic feedback echoes from microphone–loudspeaker systems.

14.1 Introduction: Acoustic and Hybrid Echoes

Echo can severely affect the quality and intelligibility of voice conversation in a telephone system. The perceived effect of an echo depends on its amplitude and time delay. In general, echoes with an appreciable amplitude and a delay of more than 1 ms are noticeable. Provided the round-trip delay is on the order of a few milliseconds, echo gives a telephone call a 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. Hence echo cancellation is an important aspect of the design of modern telecommunication systems such as conventional wireline telephones, hands-free phones, cellular mobile (wireless) phones, or teleconference systems. There are two types of echo in a telephone system (Figure 14.1):

(a) acoustic echo due to acoustic coupling between the speaker and the microphone in hands-free phones, mobile phones and teleconference systems;
(b) electrical line echo due to mismatch at the hybrid circuit connecting a 2-wire subscriber line to a 4-wire truck line in the public switched telephone network.

In the early days of expansion of telephone networks, the cost of running a 4-wire line from the local exchange to subscribers' premises was considered uneconomical. Hence, at the exchange the 4-wire truck lines are converted to 2-wire subscribers local lines using a 2/4-wire hybrid bridge circuit. At the receiver due to any imbalance between the 4/2-wire bridge circuit, some of the signal energy of the 4-wire circuit is bounced back

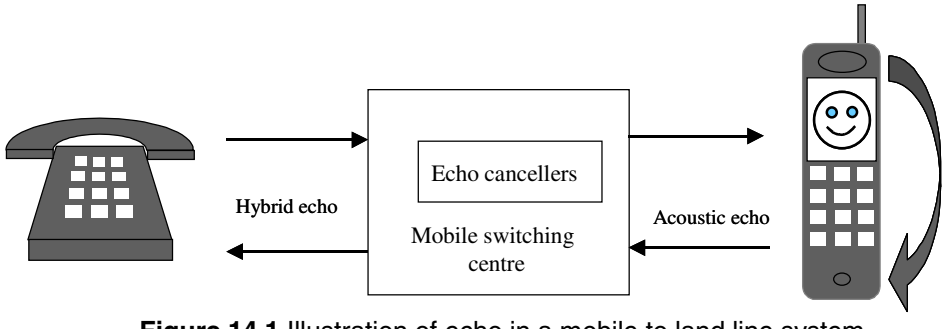


Figure 14.1 Illustration of echo in a mobile to land line system.

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 voice signals are processed at two points in the network: first voice signals are digitised, compressed and coded within the mobile handset, and then 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. A 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.

14.2 Telephone Line Hybrid Echo

Hybrid echo is the main source of echo generated from the public-switched telephone network (PSTN). Echoes on a telephone line are due to the reflection of signals at the points of impedance mismatch on the connecting circuits. Conventionally, telephones in a given geographical area are connected to an exchange by a 2-wire twisted line, called the subscriber's lineline, which serves to receive and transmit signals. 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 2-wire local lines or because the

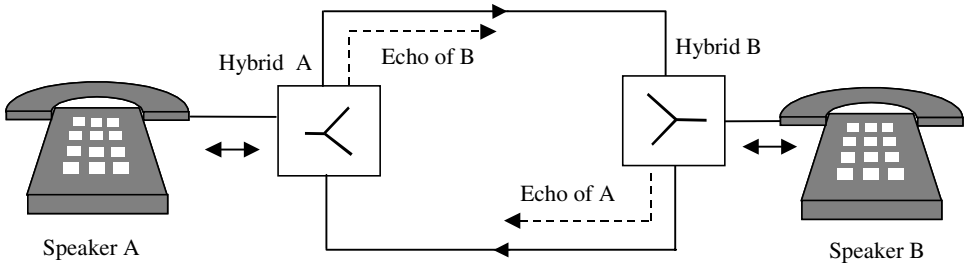


Figure 14.2 Illustration of a telephone call set up by connection of 2-wire subscriber's via hybrids to 4-wire lines at the exchange.

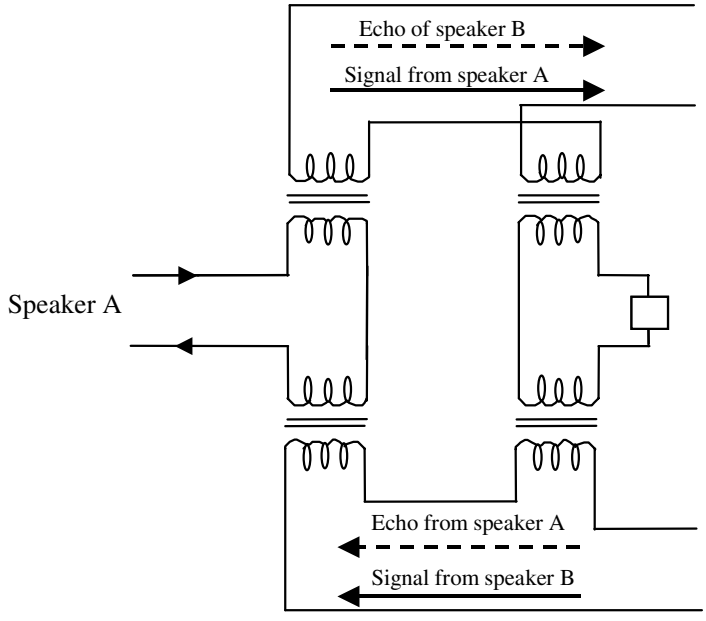


Figure 14.3 A 2-wire to 4-wire hybrid circuit.

distances are relatively small and the resulting low-delay echoes 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 2-wire telephone line is required for each direction of transmission.

To establish a long-distance call, at each end, a 2-wire subscriber's line must be connected to a 4-wire line at the exchange, as illustrated in Figure 14.2. The device that connects the 2-wire subscriber's loop to the 4-wire line is called a hybrid, and is shown in Figure 14.3. As shown the hybrid is basically a three-port bridge circuit. If the hybrid bridge were perfectly balanced then there would be no reflection or 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 4-wire lines becomes coupled back onto itself and produces an echo. Echo is often measured in terms of the echo return loss (ERL); the higher the echo return loss the lower will be the echo. Telephone line echoes are undesirable, and become annoying when the echo amplitude is relatively high and the echo delay is long. For example when a long-distance call is

made via a satellite the round-trip echo delay can be as long as 600 ms, and echoes can become disruptive. Also, as already mentioned, there are appreciable delays of up to 200 ms inherent in digital mobile phones, which make any echo quite noticeable. For this reason the employment of echo cancellers in mobile switching centres is mandatory.

14.3 Hybrid Echo Suppression

The development of echo reduction began in the late 1950s with the advent of echo suppression systems. Echo suppressors were first employed to manage the echo generated primarily in satellite circuits. An echo suppresser (Figure 14.4) 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 the speaker B to the speaker A, or the echo of speaker A. If the echo detector decides that the signal is an echo then the signal is heavily attenuated. There is a similar echo suppression unit from speaker A to speaker B. The performance of an echo suppresser depends on the accuracy of the echo/speech classification subsystem. Echo of speech often has a smaller amplitude level than the speech signal, but

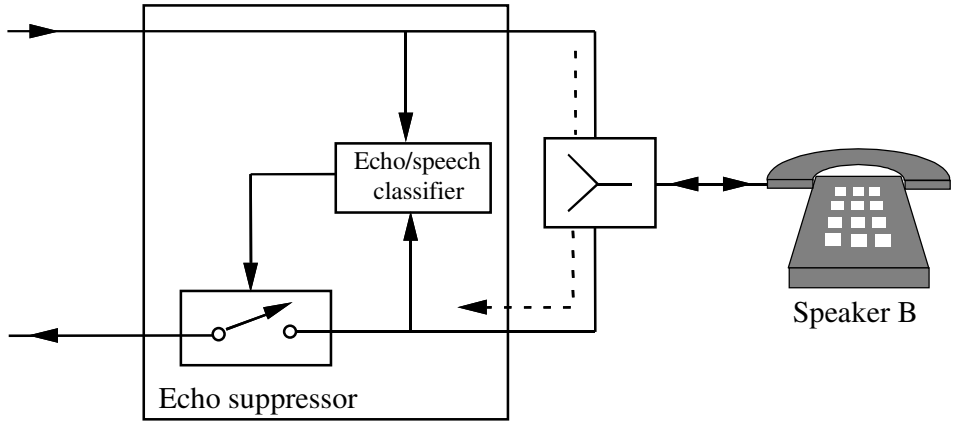


Figure 14.4 Block diagram illustration of an echo suppression system.

otherwise it has mainly the same spectral characteristics and statistics as those of the speech. Therefore the only basis for discrimination of speech from echo is the signal level. As a result, the speech/echo classifier may wrongly classify and let through high-level echoes as speech, or attenuate low-level speech as echo. For terrestrial circuits, echo suppressers have been well designed, with an acceptable level of false decisions and a good performance. The performance of an echo suppresser depends on the time delay of the echo. In general, echo suppressers perform well when the round-trip delay of the echo is less than 100 ms. For a conversation routed via a geostationary satellite the round-trip delay may be as much as 600 ms. Such long delays can change the pattern of conversation and result in a significant increase in speech/echo classification errors. When the delay is long, echo suppressers fail to perform satisfactorily, and this results in choppy first syllables and artificial volume adjustment. A system that is effective with both short and long time delays is the adaptive echo canceller introduced next.

14.4 Adaptive Echo Cancellation

Echo cancellation was developed in the early 1960s by AT&T Bell 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 14.5 illustrates the operation of an adaptive line echo canceller. The speech signal on the line from speaker A to speaker B is input to the 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 and synthesis 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.

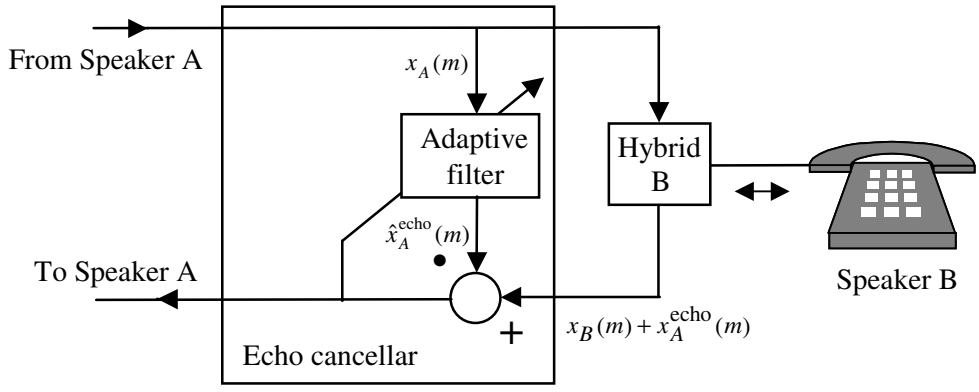


Figure 14.5 Block diagram illustration of an adaptive echo cancellation system.

Assuming that the signal on the line from speaker B to speaker A, $y_B(m)$, is composed of the speech of speaker B, $x_B(m)$, plus the echo of speaker A, $x_A^{\text{echo}}(m)$, we have

$$y_B(m) = x_B(m) + x_A^{\text{echo}}(m) \tag{14.1}$$

In practice, speech and echo signals are not simultaneously present on a phone line. This, as pointed out shortly, can be used to simplify the adaptation process. Assuming that the echo synthesiser is an FIR filter, the filter output estimate of the echo signal can be expressed as

$$\hat{x}_A^{\text{echo}}(m) = \sum_{k=0}^{P-1} w_k(m)\, x_A(m-k) \tag{14.2}$$

where $w_k(m)$ are the time-varying coefficients of an adaptive FIR filter and $\hat{x}_A^{\text{echo}}(m)$ is an estimate of the echo of speaker A on the line from speaker B to speaker A. The residual echo signal, or the error signal, after echo subtraction is given by

$$\begin{aligned} e(m) &= y_B(m) - \hat{x}_A^{\text{echo}}(m) \\ &= x_B(m) + x_A^{\text{echo}}(m) - \sum_{k=0}^{P-1} w_k(m)\, x_A(m-k) \end{aligned} \tag{14.3}$$

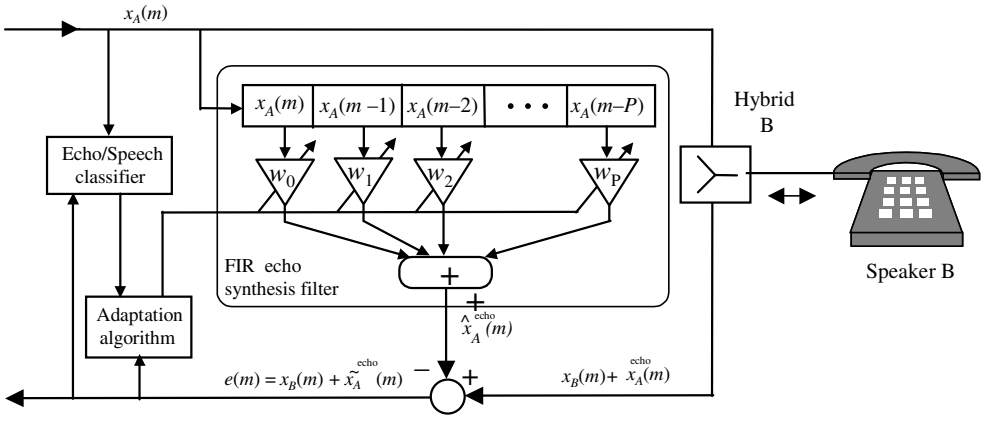


Figure 14.6 Illustration of an echo canceller using an adaptive FIR filter and incorporation a echo/speech classifier.

For those time instants when speaker A is talking, and speaker B is listening and silent, and only echo is present from line B to A, we have

$$\begin{aligned} e(m) = \tilde{x}_A^{\text{echo}}(m) &= x_A^{\text{echo}}(m) - \hat{x}_A^{\text{echo}}(m) \\ &= x_A^{\text{echo}}(m) - \sum_{k=0}^{P-1} w_k(m)\, x_A(m-k) \end{aligned} \tag{14.4}$$

where $\tilde{x}_A^{\text{echo}}(m)$ is the residual echo. An echo canceller using an adaptive FIR filter is illustrated in Figure 14.6. The magnitude of the residual echo depends on the ability of the echo canceller to synthesise a replica of the echo, and this in turn depends on the adaptation algorithm discussed next.

14.4.1 Echo Canceller Adaptation Methods

The echo canceller coefficients $w_k(m)$ 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 $x_A(m)$ and $x_B(m)$ are uncorrelated, the energy on the telephone line from B to A is minimised when the echo canceller output $\hat{x}_A^{\text{echo}}(m)$ is equal to the echo $x_A^{\text{echo}}(m)$ on the line. The echo canceller coefficients may be adapted using one of the variants of the recursive least square error (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

$$\boldsymbol{w}(m) = \boldsymbol{w}(m-1) + \mu \frac{e(m)}{\boldsymbol{x}(m)_A^{\mathrm{T}} \boldsymbol{x}_A(m)} \boldsymbol{x}_A(m) \tag{14.5}$$

where $\boldsymbol{x}_A(m)=[x_A(m), ..., x_A(m-P)]$ and $\boldsymbol{w}(m)=[w_0(m), ..., w_{P-1}(m)]$ are the input signal vector and the coefficient vector of the echo canceller, and $e(m)$ is the difference between the signal on the echo line and the output of the echo synthesiser. Note that the normalising quantity $\boldsymbol{x}(m)_A^{\mathrm{T}} \boldsymbol{x}_A(m)$ is the energy of the input speech to the adaptive filter. The scalar μ is the adaptation step size, and controls the speed of convergence, the steady-state error and the stability of the adaptation process.

14.4.2 Convergence of Line Echo Canceller

For satisfactory performance, the echo canceller should have a fast convergence rate, so that it can adequately track changes in the telephone line and the signal characteristics. The convergence of an echo canceller is affected by the following factors:

(a) *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.

(b) *Simultaneous conversation.* In a telephone conversation, usually the talkers do not speak simultaneously, and hence speech and echo are seldom present on a line at the same time. This observation simplifies the echo cancellation problem and substantially aids the correct functioning of adaptive echo cancellers. Problems arise during the periods when both speakers talk at the same time. This is because speech and its echo have

similar characteristics and occupy basically the same bandwidth. When the reference signal contains both echo and speech, the adaptation process can lose track, and the echo cancellation process can attempt to cancel out and distort the speech signal. One method of avoiding this problem is to use a speech activity detector, and freeze the adaptation process during periods when speech and echo are simultaneously present on a line, as shown in Figure 14.6. In this system, the effect of a speech/echo misclassification is that the echo may not be optimally cancelled out. This is more acceptable than is the case in echo suppressors, where the effect of a misclassification is the suppression and loss of a part of the speech.

(c) *The adaptation algorithm.* Most echo cancellers use variants of the LMS adaptation algorithm. The attractions of the LMS are its relatively low memory and computational requirements and its ease of implementation and monitoring. The main drawback of LMS is that it can be sensitive to the eigenvalue spread of the input signal and is not particularly fast in its convergence rate. However, in practice, LMS adaptation has produced effective line echo cancellation systems. The recursive least square (RLS) error methods have a faster convergence rate and a better minimum mean square error performance. With the increasing availability of low-cost high-speed dedicated DSP processors, implementation of higher-performance and computationally intensive echo cancellers based on RLS are now feasible.

14.4.3 Echo Cancellation for Digital Data Transmission

Echo cancellation becomes more complex with the increasing integration of wireline telephone systems and mobile cellular systems, and the use of digital transmission methods such as asynchronous transfer mode (ATM) for integrated transmission of data, image and voice. For example, in ATM based systems, the voice transmission delay varies depending on the route taken by the cells that carry the voice signals. This variable delay added to the delay inherent in digital voice coding complicates the echo cancellation process.

The 2-wire subscriber telephone lines that were originally intended to carry relatively low-bandwidth voice signals are now used to provide telephone users with high-speed digital data links and digital services such as video-on-demand and internet services using digital transmission

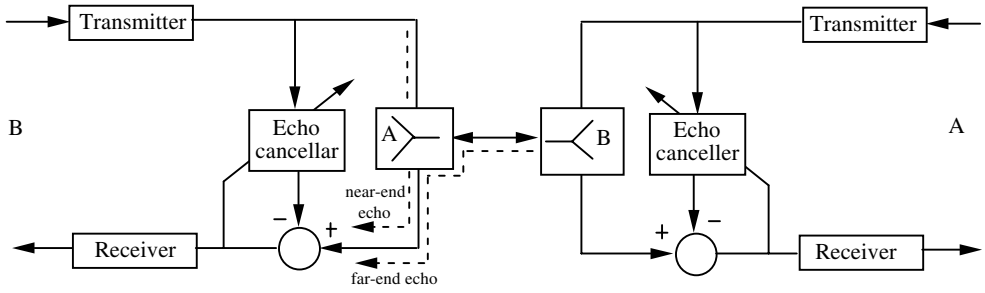


Figure 14.7 Echo cancellation in digital modems using 2-wire subscriber's loop.

methods such as the asynchronous digital subscriber line (ADSL). Traditionally, the bandwidth of the subscribers line is limited by low-pass filters at the core network to 3.4 kHz. Within this bandwidth, voice-band modems can provide data rates of around 30 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 14.7 shows a system for providing a full-duplex digital service over a 2-wire subscriber's loop. To provide simultaneous transmission of data in both directions within the same bandwidth over the subscriber's line, echo cancellation is needed. The echoes on a line consist of the near-end echo which loops back at the first or the near hybrid, and the far-end echo which is the signal that loops back at a hybrid some distance away. The main purpose of the echo canceller is to cancel the near-end echo. Since the digital signal coming from a far-end may be attenuated by 40–50 dB, the near echo on a high speed data transmission line can be as much as 40–50 dB above the desired signal level. For reliable data communication the echo canceller must provide 50–60 dB attenuation of the echo signal so that the signal power remains at 10 dB above the echo.

14.5 Acoustic Echo

Acoustic echo results from a feedback path set up between the speaker and the microphone in a mobile phone, hands-free phone, teleconference or hearing aid system. Acoustic echo is usually reflected from a multitude of different surfaces, such as walls, ceilings and floors, and travels through

different paths. If the time delay is not too long then the acoustic echo may be perceived as a soft reverberation, and may add to the artistic quality of the sound. Concert halls and church halls with desirable reverberation characteristics can enhance the quality of a musical performance. However, acoustic echo is a well-known problem with hands-free telephones, teleconference systems, public address systems, mobile phones, and hearing aids, and is due to acoustic feedback coupling of sound waves between the loudspeakers and microphones. Acoustic echo can result from a combination of direct acoustic coupling and multipath effect where the sound wave is reflected from various surfaces and then picked up by the microphone. In its worst case, acoustic feedback can result in howling if a significant proportion of the sound energy transmitted by the loudspeaker is received back at the microphone and circulated in the feedback loop. The overall round gain of an acoustic feedback loop depends on the frequency responses of the electrical and the acoustic signal paths. The undesirable effects of the electrical sections on the acoustic feedback can be reduced by designing systems that have a flat frequency response. The main problem is in the acoustic feedback path and the reverberating characteristics of the room. If the microphone–speaker–room system is excited at a frequency whose loop gain is greater than unity then the signal is amplified each time it circulates round the loop, and feedback howling results. In practice, the howling is limited by the 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 the howling but it is not effective for removal of the overall echo of the acoustic feedback. Another approach is to reduce the feedback loop-gain at those frequencies where the acoustic feedback energy is concentrated. This may be achieved 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.

The most effective method of acoustic feedback removal is the use of an adaptive feedback cancellation system. Figure 14.8 illustrates a model of an acoustic feedback environment, comprising a microphone, a loudspeaker and the reverberating space of a room. The z-transfer function of a linear model of the acoustic feedback environment may be expressed as

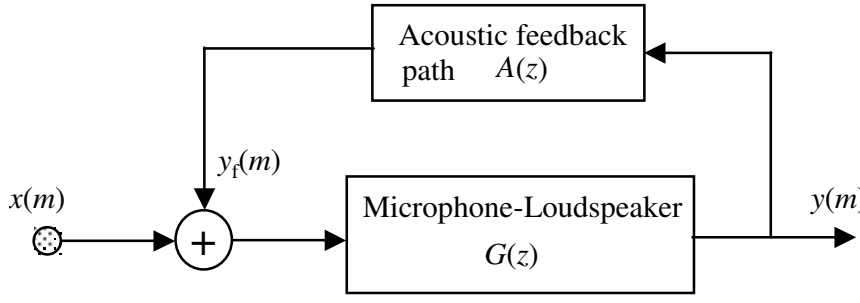


Figure 14.8 Configuration of a feedback model for a microphone–loudspeaker–room system.

$$H(z) = \frac{G(z)}{1 - G(z)A(z)} \tag{14.6}$$

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 (14.6) can be simplified to

$$H(z) = \frac{G}{1 - G\,A(z)} \tag{14.7}$$

Note that in Equation (14.6), owing to the reverberating character of the room, the acoustic feedback path $A(z)$ is itself a feedback system. The reverberating characteristics of the acoustic environment may be modelled by an all-pole linear predictive model, or alternatively a relatively long FIR model.

The equivalent time-domain input/output relation for the linear filter model of Equation (14.7) is given by the following difference equation:

$$y(m) = G\sum_{k=0}^{P-1} a_k(m)y(m-k) + Gx(m) \tag{14.8}$$

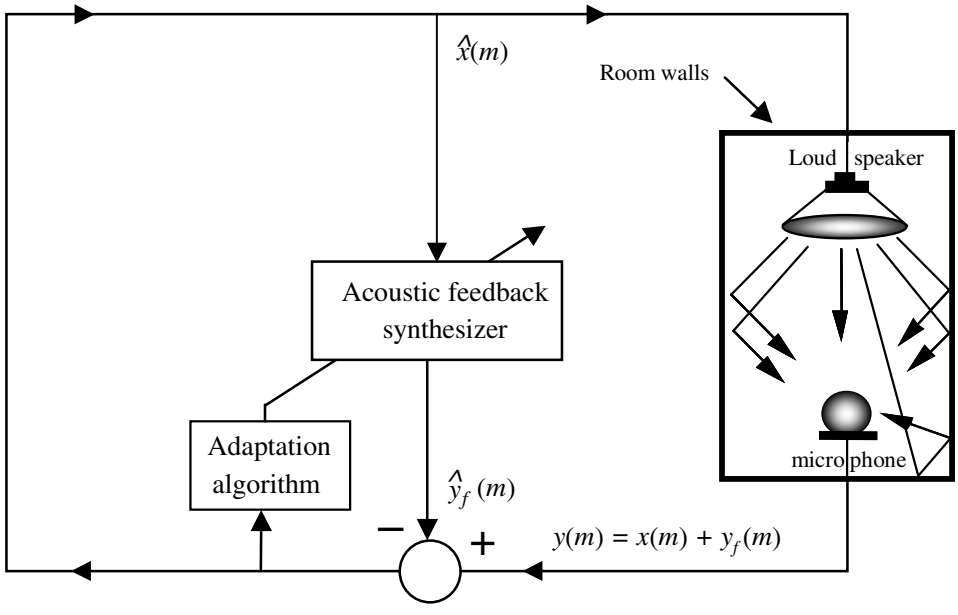


Figure 14.9 Illustration of adaptive acoustic feedback cancellation in a conference room environment.

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 14.9 is an illustration of an acoustic feedback cancellation system. In an acoustic feedback environment, the total input signal to the microphone is given as the sum of any new input to the microphone $x(m)$ plus the unwanted acoustic feedback signal $y_f(m)$:

$$y(m) = x(m) + y_f(m) \tag{14.9}$$

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

$$\hat{y}_f(m) = \sum_{k=0}^{P-1} \hat{a}_k(m) y(m-k) \tag{14.10}$$

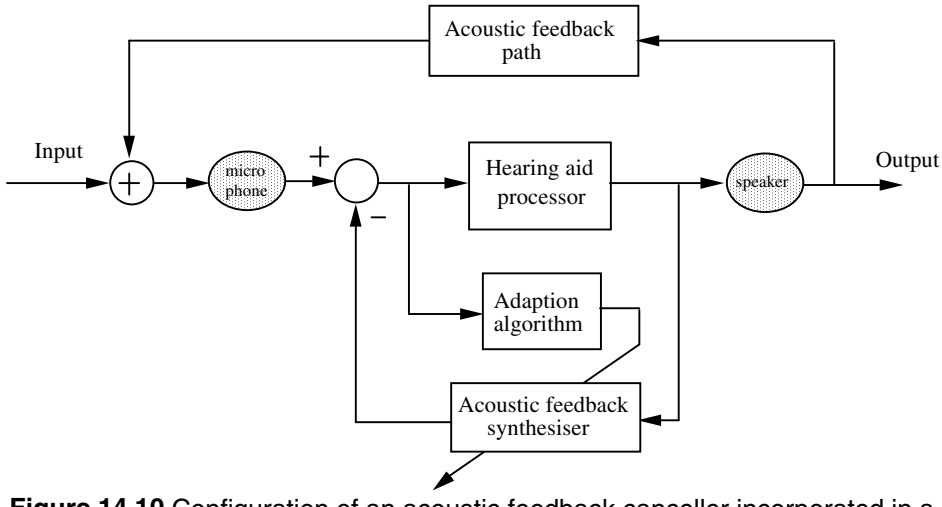


Figure 14.10 Configuration of an acoustic feedback canceller incorporated in a hearing aid system.

The filter coefficients are adapted to minimise the energy of an error signal defined as

$$e(m) = x(m) + y_f(m) - \hat{y}_f(m) \tag{14.11}$$

The adaptation criterion is usually the minimum mean square error criterion and the adaptation algorithm is a variant of the LMS or the RLS method. The problem of acoustic echo cancellation is more complex than line echo cancellation for a number of reasons. First, acoustic echo is usually much longer (up to a second) than terrestrial telephone line echoes. In fact, the delay of an acoustic echo is similar to or more than a line echo routed via a geostationary satellite system.

The large delay of an acoustic echo path implies that impractically large filters on the order of a few thousand coefficients may be required. The stable and speedy adaptation of filters of such length presents a difficult problem. Secondly, the characteristics of an acoustic echo path is more non-stationary compared with that of a telephone line echo. For example, the opening or closing of a door, or people moving in or out of a room, can suddenly change the acoustic character of a conference room. Thirdly, acoustic echoes are due to signals reflected back from a multitude of different paths, off the walls, the floor, the ceiling, the windows etc. Finally, the propagation and diffusion characteristics of the acoustic space of a room is a non-linear process, and is not well approximated by a lumped FIR (or

IIR) linear filter. In comparison, it is more reasonable to model the characteristics of a telephone line echo with a linear filter. In any case, for acoustic echo cancellation, the filter must have a large impulse response and should be able to quickly track fast changes in echo path characteristics.

An important application of acoustic feedback cancellation is in hearing aid systems. A hearing aid system can be modelled as a feedback system as shown in Figure 14.10. The maximum usable gain of a hearing aid system is limited by the acoustic feedback between the microphone and the speaker. Figure 14.10 illustrates the configuration of a feedback canceller in a hearing aid system. The acoustic feedback synthesiser has the same input as the acoustic feedback path. An adaptation algorithm adjusts the coefficients of the synthesiser to cancel out the feedback signals picked up by the microphone, before the microphone output is fed into the speaker.

14.6 Sub-Band Acoustic Echo Cancellation

In addition to the complex and varying nature of room acoustics, there are two main problems in acoustic echo cancellation. First, the echo delay is relatively long, and therefore the FIR echo synthesiser must have a large number of coefficients, say 2000 or more. Secondly, the long impulse response of the FIR filter and the large eigenvalue spread of the speech signals result in a slow, and uneven, rate of convergence of the adaptation process.

A sub-band-based echo canceller alleviates the problems associated with the required filter length and the speed of convergence. The sub-band-based system is shown in Figure 14.11. The sub-band analyser splits the input signal into N sub-bands. Assuming that the sub-bands have equal bandwidth, each sub-band occupies only $1/N$ of the baseband frequency, and can therefore be decimated (down sampled) without loss of information. For simplicity, assume that all sub-bands are down-sampled by the same factor R. The main advantages of a sub-band echo canceller are a reduction in filter length and a gain in the speed of convergence as explained below:

(a) *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.
(b) *Reduction in computational complexity*. The computational complexity of an LMS-type adaptive filter depends directly on the

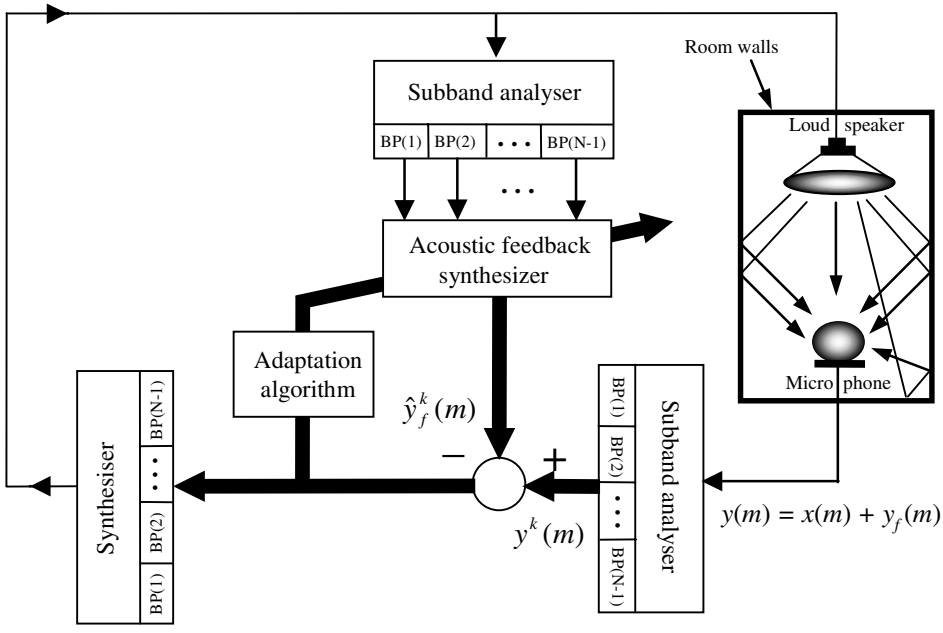


Figure 14.11 Configuration of a sub-band acoustic echo cancellation system.

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.

(c) *Speed of convergence*. The speed of convergence depends on both the filter length and the eigenvalue spread of the signal. The speed of convergence increases with the decrease in the length of the FIR filter for each sub-band. A more important factor affecting the convergence of adaptive filter is the eigenvalue spread of the autocorrelation matrix of the input signal. As the spectrum of a signal becomes flatter, the spread of its eigenvalues decreases, and the speed of convergence of the adaptive filter increases. In general, the signal within each sub-band is expected to have a flatter spectrum than the full band signal. This aids the speed of convergence. However, it must be noted that the attenuation of sub-band filters at the edges of the spectrum of each band creates some very small eigenvalues.

14.7 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 2/4-wire hybrid bridge. In Section 14.2, line echo suppression and adaptive line echo cancellation were considered. For adaptation of an echo canceller, the LMS or the RLS adaptation methods can be used. The RLS methods provides a faster convergence rate and better overall performance at the cost of higher computational complexity.

In Section 14.3, we considered the acoustic coupling between a loudspeaker and a microphone system. Acoustic feedback echo can result in howling, and can disrupt the performance of teleconference, hands-free telephones, and hearing aid systems. The main problems in implementation of acoustic echo cancellation systems are the requirement for a large filter to model the relatively long echo, and the adaptation problems associated with the eigenvalue spread of the signal. The sub-band echo canceller introduced in Section 14.4 alleviates these problems.

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**, pp. 1611–1617.

CARTER G. (1987) Coherence and Time Delay Estimation. Proc. IEEE. **75**, **2**, pp. 236–55.

FLANAGAN J.L. *et al.* (1991) Autodirective Microphone systems. Acoustica **73**, pp.58–71.

FLANAGAN J.L. *et al.* (1985) Computer-Steered Microphone Arrays for Sound Transduction in Large Rooms. J. Acoust. Soc. Amer., **78**, pp. 1508–1518.

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**, pp. 3589-3592.

GILLOIRE A. and VETTERLI M. (1994) Adaptive Filtering in Sub-bands with Critical Sampling: Analysis, Experiments and Applications to Acoustic Echo Cancellation, IEEE. Trans. Signal Processing, **40**, pp. 320–28.

GRITTON C.W. and LIN D.W. (1984) Echo Cancellation Algorithms, IEEE ASSP Mag., **1**, **2**, pp. 30–37.

GUSTAFSSON S. and MARTIN R. (1997) Combined Acoustic Echo Control and Noise Reduction for Mobile Communications, Proc. EuroSpeech-97, pp. 1403–1406.

HANSLER E. (1992) The Hands-Free Telephone Problem An Annotated Bibliography. Signal Processing, **27**, pp. 259–71.

HART J.E., NAYLOR P.A. and TANRIKULU O. (1993) Polyphase All-pass IIR Structures for Subband Acoustic Echo Cancellation. EuroSpeech-93, **3**, pp. 1813-1816.

HUA YE and BO-XIA WU (1991) A New Double-Talk Detection Algorithm Based on the Orthogonality Theorem. IEEE Trans on Communications, **39**, **11**, pp. 1542-1545, Nov.

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, pp. 2570-73.

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**, pp. 3043-46.

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, Toronto, D. Van Nostrand Inc.

SCHROEDER M.R. (1964) Improvement of Acoustic-Feedback Stability by Frequency Shifting. J. Acoust. Soc. Amer., **36**, pp. 1718-1724.

SILVERMAN H.F. *et al.* (1992) A Two-Stage Algorithm for Determining Talker Location from Linear Microphone Array Data. Computer Speech and Language, **6**, pp. 129-52.

SONDHI M.M. and BERKLEY D.A. (1980) Silencing Echoes on the Telephone Network. Proc. IEEE, **68**, pp. 948-63.

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.

SONDHI M.M. (1967) An Adaptive Echo Canceller. Bell Syst. tech. J., **46**, pp. 497-511.

TANRIKULU O., *etal.* (1995) Finite-Precision Design and Implementation of All-Pass Polyphase Networks for Echo Cancellation in subbands. IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing, ICASSP-95, **5**, pp. 3039-42.

VAIDYANATHAN P.P. (1993) Multirate Systems and Filter Banks. Prentice-Hall.

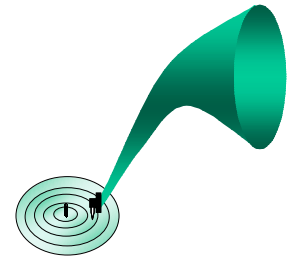
WIDROW B., MCCOOL J.M., LARIMORE M.G. and JOHNSON C.R. (1976) Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filters. Proceedings of the IEEE, **64**, **8**, pp. 1151-62.

ZELINSKI R. (1988) A Microphone Array with Adaptive Post-Filtering for Noise Reduction in Reverberant Rooms. IEEE. Proc. Int. Conf. Acoustics, Speech and Signal Processing, ICASSP-88, pp. 2578-81.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

15



CHANNEL EQUALIZATION 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 equalization for restoration of a signal distorted in transmission through a communication channel. Blind equalization has a wide range of applications, for example in digital telecommunications for removal of intersymbol 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 equalization 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 equalization 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 equalization process. This chapter begins with an introduction to the basic ideas of deconvolution and channel equalization. We study blind equalization based on the channel input power spectrum, equalization through separation of the input signal and channel response models, Bayesian equalization, nonlinear adaptive equalization for digital communication channels, and equalization of maximum-phase channels using higher-order statistics.

15.1 Introduction

In this chapter we consider the recovery of a signal distorted, in transmission through a channel, by a convolutional process and observed in additive noise. The process of recovery of a signal convolved with the impulse response of a communication channel, or a recording medium, is known as deconvolution or equalization. Figure 15.1 illustrates a typical model for a distorted and noisy signal, followed by an equalizer. 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

$$y(m)=h[x(m)]+n(m) \tag{15.1}$$

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, or a slowly time-varying, linear transversal filter. For a linear transversal filter model of the channel, Equation (15.1) becomes

$$y(m)=\sum_{k=0}^{P-1} h_k(m)x(m-k)+n(m) \tag{15.2}$$

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 (15.2) becomes

$$Y(f)=X(f)H(f)+N(f) \tag{15.3}$$

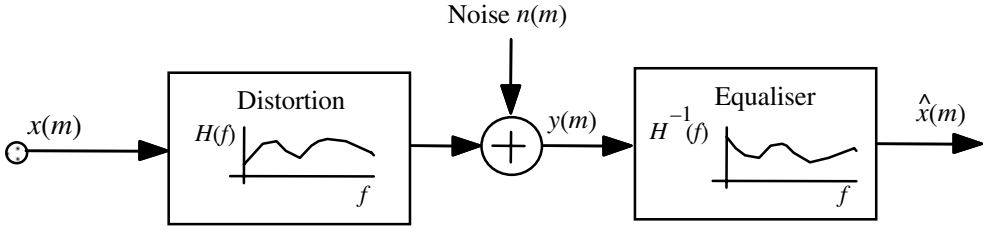


Figure 15.1 Illustration of a channel distortion model followed by an equalizer.

where $Y(f)$, $X(f)$, $H(f)$ and $N(f)$ are the frequency spectra of the channel output, the channel input, the channel response and the additive noise respectively. Ignoring the noise term and taking the logarithm of Equation (15.3) yields

$$\ln|Y(f)| = \ln|X(f)| + \ln|H(f)| \tag{15.4}$$

From Equation (15.4), in the log-frequency domain the effect of channel distortion is the addition of a "tilt" term $\ln|H(f)|$ to the signal spectrum.

15.1.1 The Ideal Inverse Channel Filter

The ideal inverse-channel filter, or the ideal equalizer, recovers the original input from the channel output signal. In the frequency domain, the ideal inverse channel filter can be expressed as

$$H(f)H^{\text{inv}}(f) = 1 \tag{15.5}$$

In Equation (15.5) $H^{\text{inv}}(f)$ is used to denote the inverse channel filter. For the ideal equalizer we have $H^{\text{inv}}(f) = H^{-1}(f)$, or, expressed in the log-frequency domain $\ln H^{\text{inv}}(f) = -\ln H(f)$. The general form of Equation (15.5) is given by the z-transform relation

$$H(z)H^{\text{inv}}(z) = z^{-N} \tag{15.6}$$

for some value of the delay N that makes the channel inversion process causal. Taking the inverse Fourier transform of Equation (15.5), we have the following convolutional relation between the impulse responses of the channel $\{h_k\}$ and the ideal inverse channel response $\{h_k^{\text{inv}}\}$:

$$\sum_k h_k^{\text{inv}} h_{i-k} = \delta(i) \tag{15.7}$$

where $\delta(i)$ is the Kronecker delta function. Assuming the channel output is noise-free and the channel is invertible, the ideal inverse channel filter can be used to reproduce the channel input signal with zero error, as follows.

The inverse filter output $\hat{x}(m)$, with the distorted signal $y(m)$ as the input, is given as

$$\begin{aligned}\hat{x}(m) &= \sum_k h_k^{\text{inv}} y(m-k) \\ &= \sum_k h_k^{\text{inv}} \sum_j h_j x(m-k-j) \\ &= \sum_i x(m-i) \sum_k h_k^{inv} h_{i-k}\end{aligned} \tag{15.8}$$

The last line of Equation (15.8) is derived by a change of variables $i=k+j$ in the second line and rearrangement of the terms. For the ideal inverse channel filter, substitution of Equation (15.7) in Equation (15.8) yields

$$\hat{x}(m) = \sum_i \delta(i) x(m-i) = x(m) \tag{15.9}$$

which is the desired result. In practice, it is not advisable to implement $H^{\text{inv}}(f)$ simply as $H^{-1}(f)$ because, in general, a channel response may be 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.

15.1.2 Equalization Error, Convolutional Noise

The equalization error signal, also called the convolutional noise, is defined as the difference between the channel equalizer output and the desired signal:

$$\begin{aligned}v(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k)\end{aligned} \tag{15.10}$$

where $\hat{h}_k^{\text{inv}}$ is an estimate of the inverse channel filter. Assuming that there is an ideal equalizer h_k^{inv} that can recover the channel input signal $x(m)$ from the channel output $y(m)$, we have

$$x(m) = \sum_{k=0}^{P-1} h_k^{\text{inv}}\, y(m-k) \tag{15.11}$$

Substitution of Equation (15.11) in Equation (15.10) yields

$$\begin{aligned} v(m) &= \sum_{k=0}^{P-1} h_k^{\text{inv}}\, y(m-k) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}}\, y(m-k) \\ &= \sum_{k=0}^{P-1} \tilde{h}_k^{\text{inv}}\, y(m-k) \end{aligned} \tag{15.12}$$

where $\tilde{h}_k^{\text{inv}} = h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}$. The equalization 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 equalization process is proceeding well, such that $\hat{x}(m)$ is a good estimate of the channel input $x(m)$, then the convolutional noise is relatively small and decorrelated and can be modelled as a zero mean Gaussian random process.

15.1.3 Blind Equalization

The equalization 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 equalizer-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 equalizer as illustrated in Figure 15.2(a). The obvious drawback of using training periods for channel equalization is that power, time and bandwidth are consumed for the equalization process.

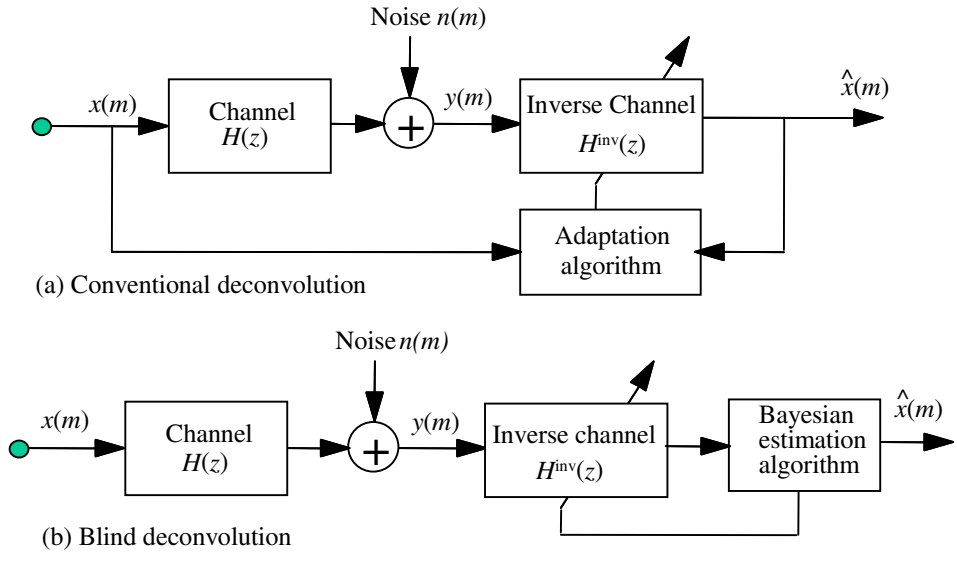


Figure 15.2 A comparative illustration of (a) a conventional equalizer with access to channel input and output, and (b) a blind equalizer.

It is preferable to have a "blind" equalization scheme that can operate without access to the channel input, as illustrated in Figure 15.2(b). Furthermore, in some applications, such as the restoration of acoustic recordings, or blurred images, all that is available is the distorted signal and the only restoration method applicable is blind equalization.

Blind equalization is feasible only if some statistical knowledge of the channel input, and perhaps that of the channel, is available. Blind equalization involves two stages of channel identification, and deconvolution of the input signal and the channel response, as follows:

(a) *Channel identification.* The general form of a channel estimator can be expressed as

$$\hat{\boldsymbol{h}} = \psi(\boldsymbol{y}, \mathcal{M}_{\boldsymbol{x}}, \mathcal{M}_{\boldsymbol{h}}) \tag{15.13}$$

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 $\mathcal{M}_{\boldsymbol{x}}$ and $\mathcal{M}_{\boldsymbol{h}}$ 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:

(i) The distribution of the channel input signal: for example, in decision-directed channel equalization, described in Section 15.5, the knowledge that the input is a binary signal is used in a binary decision device to estimate the channel input and to "direct" the equalizer adaptation process.

(ii) the relative durations of the channel input and the channel impulse response: the duration of a channel impulse response is usually orders of magnitude smaller than that of the channel input. This observation is used in Section 15.3.1 to estimate a stationary channel from the long-time averages of the channel output.

(iii) The stationary, or time-varying characteristics of the input signal process and the channel: in Section 15.3.1, a method is described for the recovery of a non-stationary signal convolved with the impulse response of a stationary channel.

(b) *Channel equalization.* Assuming that the channel is invertible, the channel input signal $x(m)$ can be recovered using an inverse channel filter as

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{15.14}$$

In the frequency domain, Equation (15.14) becomes

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f) Y(f) \tag{15.15}$$

In practice, perfect recovery of the channel input may not be possible, either because the channel is non-invertible or because the output is observed in noise. A channel is non-invertible if:

(i) 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 following section.

(ii) The channel transfer function maps many inputs to the same output: in these situations, a stable closed-form equation for the inverse channel does not exist, and instead an iterative deconvolution method is used. Figure 15.3 illustrates the frequency response of a channel that has one invertible and two 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.

15.1.4 Minimum- and Maximum-Phase Channels

For stability, all the poles of the transfer function of a channel must lie inside the unit circle. If all the zeros of the transfer function are also inside the unit circle then the channel is said to be a minimum-phase channel. If some of the zeros are outside the unit circle then the 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

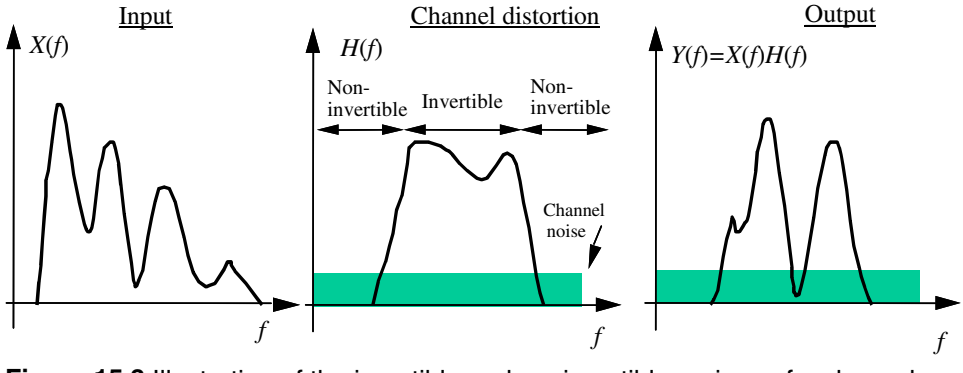


Figure 15.3 Illustration of the invertible and noninvertible regions of a channel.

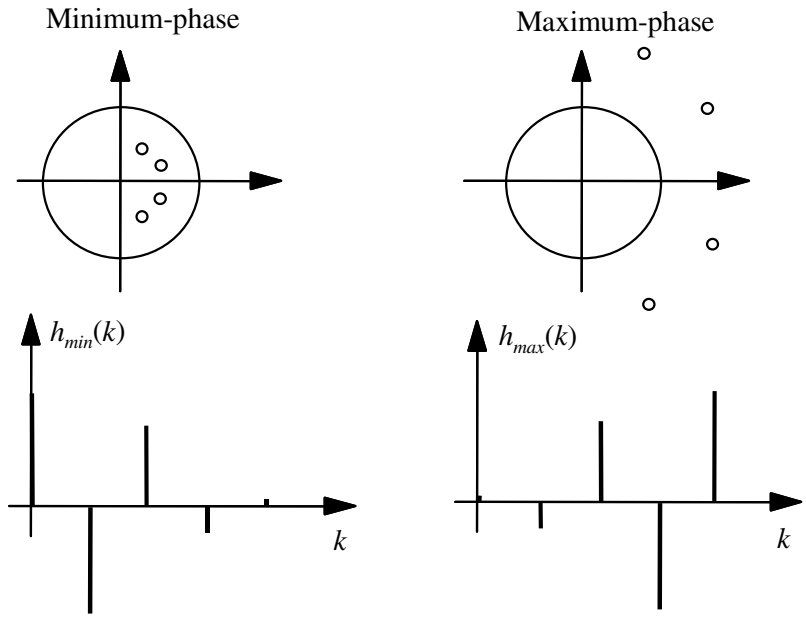


Figure 15.4 Illustration of the zero diagram and impulse response of fourth order maximum-phase and minimum-phase FIR filters.

channel may be obtained by truncating the impulse response of the inverse filter. Figure 15.3 illustrates examples of maximum-phase and minimum-phase fourth-order FIR filters.

When both the channel input and output signals are available, in the correct synchrony, it is possible to estimate the channel magnitude and phase response using the conventional least square error criterion. In blind deconvolution, there is no access to the exact instantaneous value or the timing of the channel input signal. The only information available is the channel output and some statistics of the channel input. The second order statistics of a signal (i.e. the correlation or the power spectrum) do not include the phase information; hence it is not possible to estimate the channel phase from the second-order statistics. Furthermore, the channel phase cannot be recovered if the input signal is Gaussian, because a Gaussian process of known mean is entirely specified by the autocovariance matrix, and autocovariance matrices do not include any phase information. For estimation of the phase of a channel, we can either use a non-linear estimate of the desired signal to direct the adaptation of a channel equalizer as in Section 15.5, or we can use the higher-order statistics as in Section 15.6.

15.1.5 Wiener Equalizer

In this section, we consider the least squared error Wiener equalization. Note that, in its conventional form, Wiener equalization is not a form of blind equalization, because the implementation of a Wiener equalizer requires the cross-correlation of the channel input and output signals, which are not available in a blind equalization application. The Wiener filter estimate of the channel input signal is given by

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{15.16}$$

where $\hat{h}_k^{\text{inv}}$ is an FIR Wiener filter estimate of the inverse channel impulse response. The equalization error signal $v(m)$ is defined as

$$v(m) = x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} y(m-k) \tag{15.17}$$

The Wiener equalizer with input $y(m)$ and desired output $x(m)$ is obtained from Equation (6.10) in Chapter 6 as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{xy} \tag{15.18}$$

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 (15.18) can be obtained by writing the noisy channel output signal in vector equation form as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{4.19}$$

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 (15.19) as

$$\boldsymbol{R}_{yy} = \mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}] = \boldsymbol{H}\boldsymbol{R}_{xx}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}_{nn} \tag{15.20}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. The cross-correlation vector $\boldsymbol{r}_{xy}$ of the channel input and output signals becomes

$$\boldsymbol{r}_{xy} = \mathcal{E}[x\boldsymbol{y}] = \boldsymbol{H}\boldsymbol{r}_{xx} \tag{15.21}$$

Substitution of Equation (15.20) and (15.21) in (15.18) yields the Wiener equalizer as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \left(\boldsymbol{H}\boldsymbol{R}_{xx}\boldsymbol{H}^{\text{T}} + \boldsymbol{R}_{nn}\right)^{-1}\boldsymbol{H}\boldsymbol{r}_{xx} \tag{15.22}$$

The derivation of the Wiener equalizer in the frequency domain is as follows. The Fourier transform of the equalizer output is given by

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f)Y(f) \tag{15.23}$$

where $Y(f)$, the channel output and $\hat{H}^{\text{inv}}(f)$ is the frequency response of the Wiener equalizer. The error signal $V(f)$ is defined as

$$\begin{aligned} V(f) &= X(f) - \hat{X}(f) \\ &= X(f) - \hat{H}^{\text{inv}}(f)Y(f) \end{aligned} \tag{15.24}$$

As in Section 6.5 minimisation of the expectation of the squared magnitude of $V(f)$ results in the frequency Wiener equalizer given by

$$\begin{aligned} \hat{H}^{\text{inv}}(f) &= \frac{P_{XY}(f)}{P_{YY}(f)} \\ &= \frac{P_{XX}(f)H^{*}(f)}{P_{XX}(f)\left|H(f)\right|^{2} + P_{NN}(f)} \end{aligned} \tag{15.25}$$

where $P_{XX}(f)$ is the channel input power spectrum, $P_{NN}(f)$ is the noise power spectrum, $P_{XY}(f)$ is the cross-power spectrum of the channel input and output signals, and $H(f)$ is the frequency response of the channel. Note that in the absence of noise, $P_{NN}(f)=0$ and the Wiener inverse filter becomes $H^{\text{inv}}(f) = H^{-1}(f)$.

15.2 Blind Equalization 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 15.5, had a bandwidth of about 200 Hz to 4 kHz. However, the limited bandwidth, or even the additive noise or scratch noise pulses, are not considered 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 (15.2), reproduced here for convenience

$$y(m) = \sum_{k=0}^{P-1} h_k x(m-k) + n(m) \tag{15.26}$$

or in the frequency domain as

$$Y(f) = X(f)H(f) + N(f) \tag{15.27}$$

where $H(f)$ is the frequency response of a linear time-invariant model of the acoustic recording instrument, and $N(f)$ is an additive noise. Multiplying

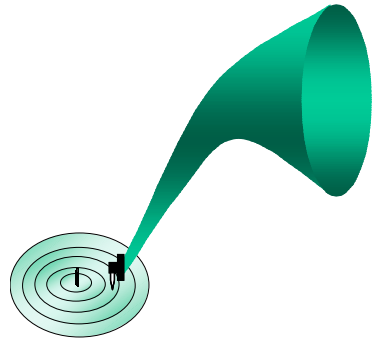


Figure 15.5 Illustration of the early acoustic recording process on a wax disc. Acoustic recordings were made by focusing the sound energy, through a horn via a sound box, diaphragm and stylus mechanism, onto a wax disc. The sound was distorted by reverberations of the horn.

both sides of Equation (15.27) with their complex conjugates, and taking the expectation, we obtain

$$\mathcal{E}[Y(f)Y^*(f)]=\mathcal{E}\left[\left(X(f)H(f)+N(f)\right)\left(X(f)H(f)+N(f)\right)^*\right] \quad (15.28)$$

Assuming the signal $X(f)$ and the noise $N(f)$ are uncorrelated Equation (15.28) becomes

$$P_{YY}(f)=P_{XX}(f)\left|H(f)\right|^2+P_{NN}(f) \quad (15.29)$$

where $P_{YY}(f)$, $P_{XX}(f)$ and $P_{NN}(f)$ are the power spectra of the distorted signal, the original signal and the noise respectively. From Equation (15.29) an estimate of the spectrum of the channel response can be obtained as

$$\left|H(f)\right|^2=\frac{P_{YY}(f)-P_{NN}(f)}{P_{XX}(f)} \quad (15.30)$$

In practice, Equation (15.30) is implemented using time-averaged estimates of the of the power spectra.

15.2.1 Homomorphic Equalization

In homomorphic equalization, the convolutional distortion is transformed, first into a multiplicative distortion through a Fourier transform of the distorted signal, and then into an additive distortion by taking the logarithm of the spectrum of the distorted signal. A further inverse Fourier transform operation converts the log-frequency variables into cepstral variables as illustrated in Figure 15.6. Through homomorphic transformation convolution becomes addition, and equalization becomes subtraction.

Ignoring the additive noise term and transforming both sides of Equation (15.27) into log-spectral variables yields

$$\ln Y(f)=\ln X(f)+\ln H(f) \quad (15.31)$$

Note that in the log-frequency domain, the effect of channel distortion is the addition of a tilt to the spectrum of the channel input. Taking the expectation of Equation (15.31) yields

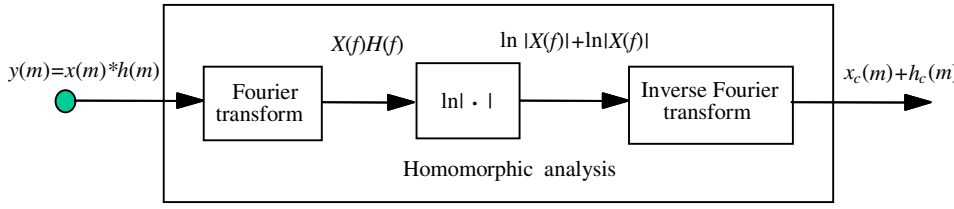


Figure 15.6 Illustration of homomorphic analysis in deconvolution.

$$\mathcal{E}[\ln Y(f)] = \mathcal{E}[\ln X(f)] + \ln H(f) \tag{15.32}$$

In Equation (15.32), it is assumed that the channel is time-invariant; hence $\mathcal{E}[\ln H(f)] = \ln H(f)$. Using the relation $\ln z = \ln|z| + j\angle z$, the term $\mathcal{E}[\ln X(f)]$ can be expressed as

$$\mathcal{E}[\ln X(f)] = \mathcal{E}[\ln|X(f)|] + j\mathcal{E}[\angle X(f)] \tag{15.33}$$

The first term on the right-hand side of Equation (15.33), $\mathcal{E}[\ln|X(f)|]$, 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 (15.32), the log-frequency spectrum of the channel can be estimated as

$$\ln H(f) = \mathcal{E}[\ln Y(f)] - \mathcal{E}[\ln X(f)] \tag{15.34}$$

In practice, when only a single record of a signal is available, the signal is divided into a number of segments, and the average signal spectrum is obtained over time across the segments. Assuming that the length of each segment is long compared with the duration of the channel impulse response, we can write an approximate convolutional relation for the i^{th} signal segment as

$$y_i(m) \approx x_i(m) * h_i(m) \tag{15.35}$$

The segments are windowed, using a Hamming or a Hanning window, to reduce the spectral leakage due to end effects at the edges of the segment. Taking the complex logarithm of the Fourier transform of Equation (15.35) yields

$$\ln Y_i(f) = \ln X_i(f) + \ln H_i(f) \tag{15.36}$$

Taking the time averages over N segments of the distorted signal record yields

$$\frac{1}{N}\sum_{i=0}^{N-1}\ln Y_i(f)=\frac{1}{N}\sum_{i=0}^{N-1}\ln X_i(f)+\frac{1}{N}\sum_{i=0}^{N-1}\ln H_i(f) \tag{15.37}$$

Estimation of the channel response from Equation (15.37) requires the average log spectrum of the undistorted signal $X(f)$. In Stockham's method for restoration of acoustic records, the expectation of the signal spectrum is obtained from a modern recording of the same musical material as that of the acoustic recording. From Equation (15.37), the estimate of the logarithm of the channel is given by

$$\ln \hat{H}(f) = \frac{1}{N}\sum_{i=0}^{N-1}\ln Y_i(f) - \frac{1}{N}\sum_{i=0}^{N-1}\ln X_i^{\mathcal{M}}(f) \tag{15.38}$$

where $X^{\mathcal{M}}(f)$ is the spectrum of a modern recording. The equalizer can then be defined as

$$\ln H^{\text{inv}}(f)=\begin{cases}-\ln \hat{H}(f), & 200\ \text{Hz} \le f \le 4000\ \text{Hz}\\ -40\ \text{dB}, & \text{otherwise}\end{cases} \tag{15.39}$$

In Equation (15.39), 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.

15.2.2 Homomorphic Equalization 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

$$\ln Y_t(f)=\ln X_t(f)+\ln H_t(f) \tag{15.40}$$

where $Y_t(f)$ and $X_t(f)$ are the channel input and output derived from a Fourier transform of the t^{th} signal segment. From Equation (15.40), the effect of a

time-invariant channel is to add a constant term ln $H(f)$ to each frequency component of the channel input $X_t(f)$, and the overall result is a time-invariant tilt of the log-frequency spectrum of the original signal. This observation suggests the use of a bank of 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

$$\ln \hat{X}_t(f) = \alpha \ln \hat{X}_{t-1}(f) + \ln Y_t(f) - \ln Y_{t-1}(f) \tag{15.41}$$

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.

15.3 Equalization 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

$$X(z) = E(z)A(z) \tag{15.42}$$

where $X(z)$ is the z-transform of the channel input signal, $A(z)$ is the z-transfer function of a linear predictive model of the channel input and $E(z)$ is the z-transform of a random excitation signal. Similarly, the channel output can be modelled by a linear predictive model $H(z)$ with input $X(z)$ and output $Y(z)$ as

$$Y(z) = X(z)H(z) \tag{15.43}$$

Figure 15.7 illustrates a cascade linear prediction model for a channel input process $X(z)$ and a channel response $H(z)$. The channel output can be expressed as

$$\begin{aligned} Y(z) &= E(z)A(z)H(z) \\ &= E(z)D(z) \end{aligned} \tag{15.44}$$

where

$$D(z) = A(z)H(z) \tag{15.45}$$

The z-transfer function of the linear prediction models of the channel input signal and the channel can be expanded as

$$A(z) = \frac{G_1}{1 - \sum_{k=1}^{P} a_k z^{-k}} = \frac{G_1}{\prod_{k=1}^{P} (1 - \alpha_k z^{-1})} \tag{15.46}$$

$$H(z) = \frac{G_2}{1 - \sum_{k=1}^{Q} b_k z^{-k}} = \frac{G_2}{\prod_{k=1}^{Q} (1 - \beta_k z^{-1})} \tag{15.47}$$

where $\{a_k, \alpha_k\}$ and $\{b_k, \beta_k\}$ are the coefficients and the poles of the linear prediction models for the channel input signal and the channel respectively. Substitution of Equations (15.46) and (15.47) in Equation (15.45) yields the combined input-channel model as

$$D(z) = \frac{G}{1 - \sum_{k=1}^{P+Q} \mathrm{d}_k z^{-k}} = \frac{G}{\prod_{k=1}^{P+Q} (1 - \gamma_k z^{-1})} \tag{15.48}$$

The total number of poles of the combined model for the input signal and the channel is the sum of the poles of the input signal model and the channel model.

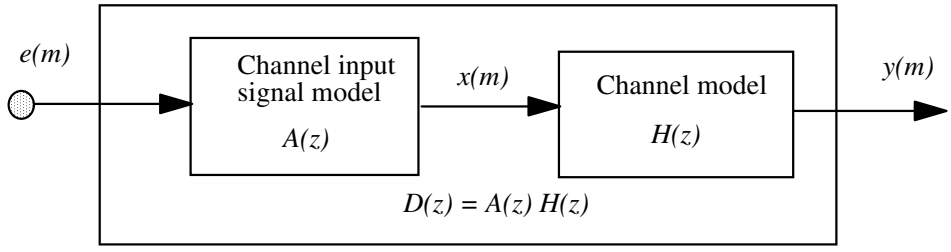


Figure 15.7 A distorted signal modelled as cascade of a signal model and a channel model.

15.3.1 Blind Equalization Through Model Factorisation

A model-based approach to blind equalization 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 factorize 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 factorization 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

$$x(m)=a_1(m)x(m-1)+a_2(m)x(m-2)+G_1(m)e(m) \tag{15.49}$$

where $a_1(m)$ and $a_2(m)$ are the time-varying coefficients of the linear predictor model, $G_1(m)$ is the input gain factor and $e(m)$ is a zero-mean, unit variance, random signal. Now let $\alpha_1(m)$ and $\alpha_2(m)$ denote the time-varying

poles of the predictor model of Equation (15.49); these poles are the roots of the polynomial

$$1 - a_1(m)z^{-1} - a_2(m)z^{-2} = \left[1 - z^{-1}\alpha_1(m)\right]\left[1 - z^{-1}\alpha_2(m)\right] = 0 \qquad (15.50)$$

Similarly, assume that the channel can be modelled by a second-order stationary linear predictive model as

$$y(m) = h_1 y(m-1) + h_2 y(m-2) + G_2 x(m) \qquad (15.51)$$

where h_1 and h_2 are the time-invariant predictor coefficients and G_2 is the channel gain. Let β_1 and β_2 denote the poles of the channel model; these are the roots of the polynomial

$$1 - h_1 z^{-1} - h_2 z^{-2} = (1 - z^{-1}\beta_1)(1 - z^{-1}\beta_2) = 0 \qquad (15.52)$$

The combined cascade of the two second-order models of Equations (15.49) and (15.51) can be written as a fourth-order linear predictive model with input $e(m)$ and output $y(m)$:

$$y(m) = d_1(m)y(m-1) + d_2(m)y(m-2) + d_3(m)y(m-3) + d_4(m)y(m-4) + Ge(m) \qquad (15.53)$$

where the combined gain $G = G_1 G_2$. The poles of the fourth order predictor model of Equation (15.53) are the roots of the following polynomial:

$$\begin{aligned} &1 - d_1(m)z^{-1} - d_2(m)z^{-2} - d_3(m)z^{-3} - d_4(m)z^{-4} = \\ &= \left[1 - z^{-1}\alpha_1(m)\right]\left[1 - z^{-1}\alpha_2(m)\right]\left[1 - z^{-1}\beta_1\right]\left[1 - z^{-1}\beta_2\right] = 0 \end{aligned} \qquad (15.54)$$

In Equation (15.54) the poles of the fourth order predictor are $\alpha_1(m)$, $\alpha_2(m)$, β_1 and β_2 . The above argument on factorisation of the poles of time-varying and stationary models can be generalised to a signal model of order P and a channel model of order Q.

In Spencer and Rayner, the separation of the stationary poles of the channel from the time-varying poles of the channel input is achieved through a clustering process. The signal record is divided into N segments and each segment is modelled by an all-pole model of order $P+Q$ where P and Q are the assumed model orders for the channel input and the channel

respectively. In all, there are $N(P+Q)$ values which are clustered to form $P+Q$ clusters. Even if both the signal and the channel were stationary, the poles extracted from different segments would have variations due to the random character of the signals from which the poles are extracted. Assuming that the variances of the estimates of the stationary poles are small compared with the variations of the time-varying poles, it is expected that, for each stationary pole of the channel, the N values extracted from N segments will form an N-point cluster of a relatively small variance. These clusters can be identified and the centre of each cluster taken as a pole of the channel model This method assumes that the poles of the time-varying signal are well separated in space from the poles of the time-invariant signal.

15.4 Bayesian Blind Deconvolution and Equalization

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 equalization 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

$$\begin{aligned}\mathcal{R}(\hat{\boldsymbol{h}} \mid \boldsymbol{y}) &= \int_{\boldsymbol{H}} \int_{\boldsymbol{X}} C(\hat{\boldsymbol{h}}, \boldsymbol{h}) f_{\boldsymbol{X},\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{h} | \boldsymbol{y})\, d\boldsymbol{x}\, d\boldsymbol{h} \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \int_{\boldsymbol{H}} C(\hat{\boldsymbol{h}}, \boldsymbol{h}) f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y} | \boldsymbol{h}) f_{\boldsymbol{H}}(\boldsymbol{h})\, d\boldsymbol{h}\end{aligned} \tag{15.55}$$

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 $\mathcal{R}(\hat{\boldsymbol{h}} \mid \boldsymbol{y})$. 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

$$\boldsymbol{y}(m)=\boldsymbol{x}(m)+\boldsymbol{h} \tag{15.56}$$

where the cepstral vectors $\boldsymbol{x}(m)$, $\boldsymbol{y}(m)$ and $\boldsymbol{h}$ are the channel input, the channel output and the channel respectively.

15.4.1 Conditional Mean Channel Estimation

A commonly used cost function in the Bayesian risk of Equation (15.55) is the mean square error $C(\boldsymbol{h}-\hat{\boldsymbol{h}})=|\boldsymbol{h}-\hat{\boldsymbol{h}}|^2$, which results in the conditional mean (CM) estimate defined as

$$\hat{\boldsymbol{h}}^{CM} = \int_{\boldsymbol{H}} \boldsymbol{h} f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h} \mid \boldsymbol{y})\, d\boldsymbol{h} \tag{15.57}$$

The posterior density of the channel input signal may be conditioned on an estimate of the channel vector $\hat{\boldsymbol{h}}$ and expressed as $f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x} \mid \boldsymbol{y},\hat{\boldsymbol{h}})$. The conditional mean of the channel input signal given the channel output $\boldsymbol{y}$ and an estimate of the channel $\hat{\boldsymbol{h}}$ is

$$\begin{aligned} \hat{\boldsymbol{x}}^{CM} &= \mathcal{E}[\boldsymbol{x} \mid \boldsymbol{y},\hat{\boldsymbol{h}}] \\ &= \int_{\boldsymbol{X}} \boldsymbol{x} f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x} \mid \boldsymbol{y},\hat{\boldsymbol{h}})\, d\boldsymbol{x} \end{aligned} \tag{15.58}$$

Equations (15.57) and (15.58) suggest a two-stage iterative method for channel estimation and the recovery of the channel input signal.

15.4.2 Maximum-Likelihood Channel Estimation

The ML channel estimate is equivalent to the case when the Bayes cost function and the channel prior are uniform. Assuming that the channel input signal has a Gaussian distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{x}}$ 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

$$f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)|\boldsymbol{h}) = \prod_{m=0}^{N-1} f_{\boldsymbol{X}}(\boldsymbol{y}(m)-\boldsymbol{h}) = \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{xx}|^{1/2}} \exp\left\{[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]^{\mathrm{T}} \boldsymbol{\Sigma}_{xx}^{-1}[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]\right\} \tag{15.59}$$

To obtain the ML estimate of the channel $\boldsymbol{h}$, the derivative of the log likelihood function $\ln f_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{h})$ with respect to $\boldsymbol{h}$ is set to zero to yield

$$\hat{\boldsymbol{h}}^{ML} = \frac{1}{N}\sum_{m=0}^{N-1}(\boldsymbol{y}(m)-\boldsymbol{\mu}_x) \tag{15.60}$$

15.4.3 Maximum A Posteriori Channel Estimation

The MAP estimate, like the ML estimate, is equivalent to a Bayesian estimator with a uniform cost function. However, the MAP estimate includes the prior pdf of the channel. The prior pdf can be used to confine the channel estimate within a desired subspace of the parameter space. Assuming that the channel input vectors are statistically independent, the posterior pdf of the channel given the observation sequence $\boldsymbol{Y}=\{\boldsymbol{y}(0), ..., \boldsymbol{y}(N-1)\}$ is

$$\begin{aligned} f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)) &= \prod_{m=0}^{N-1} \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y}(m))} f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y}(m)|\boldsymbol{h}) f_{\boldsymbol{H}}(\boldsymbol{h}) \\ &= \prod_{m=0}^{N-1} \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y}(m))} f_{\boldsymbol{X}}(\boldsymbol{y}(m)-\boldsymbol{h}) f_{\boldsymbol{H}}(\boldsymbol{h}) \end{aligned} \tag{15.61}$$

Assuming that the channel input $\boldsymbol{x}(m)$ is Gaussian, $f_{\boldsymbol{X}}(\boldsymbol{x}(m))=\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$, with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the channel $\boldsymbol{h}$ is also Gaussian, $f_{\boldsymbol{H}}(\boldsymbol{h})=\mathcal{N}(\boldsymbol{h}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_{hh})$, with mean vector $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Sigma}_{hh}$, the logarithm of the posterior pdf is

$$\ln f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)) = -\sum_{m=0}^{N-1}\ln f(\boldsymbol{y}(m)) - NP\ln(2\pi) - \frac{1}{2}\ln(|\boldsymbol{\Sigma}_{\boldsymbol{xx}}||\boldsymbol{\Sigma}_{\boldsymbol{hh}}|)$$
$$-\sum_{m=0}^{N-1}\frac{1}{2}\left\{[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x}}]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x}}]+(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})\right\} \tag{15.62}$$

The MAP channel estimate, obtained by setting the derivative of the log posterior function $\ln f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{y})$ to zero, is

$$\hat{\boldsymbol{h}}^{MAP} = (\boldsymbol{\Sigma}_{\boldsymbol{xx}}+\boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{hh}}(\bar{\boldsymbol{y}}-\boldsymbol{\mu}_{\boldsymbol{x}})+(\boldsymbol{\Sigma}_{\boldsymbol{xx}}+\boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\mu}_{\boldsymbol{h}} \tag{15.63}$$

where

$$\bar{\boldsymbol{y}} = \frac{1}{N}\sum_{m=0}^{N-1}\boldsymbol{y}(m) \tag{15.64}$$

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.

15.4.4 Channel Equalization Based on Hidden Markov Models

This section considers blind deconvolution in applications where the statistics of the channel input are modelled by a set of hidden Markov models. An application of this method, illustrated in Figure 15.8, is in recognition of speech distorted by a communication channel or a microphone. A hidden Markov model (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 of an HMM $\mathcal{M}_i$ and a sequence of N P-dimensional channel input vectors $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(N-1)]$ can be expressed in terms of the state transition and the observation pdfs of $\mathcal{M}_i$ as

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X}\mid\mathcal{M}_i)=\sum_{\mathbf{s}} f_{\boldsymbol{X}|\mathcal{M},\boldsymbol{S}}(\boldsymbol{X}\mid\mathcal{M}_i,\mathbf{s})P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}\mid\mathcal{M}_i) \tag{15.65}$$

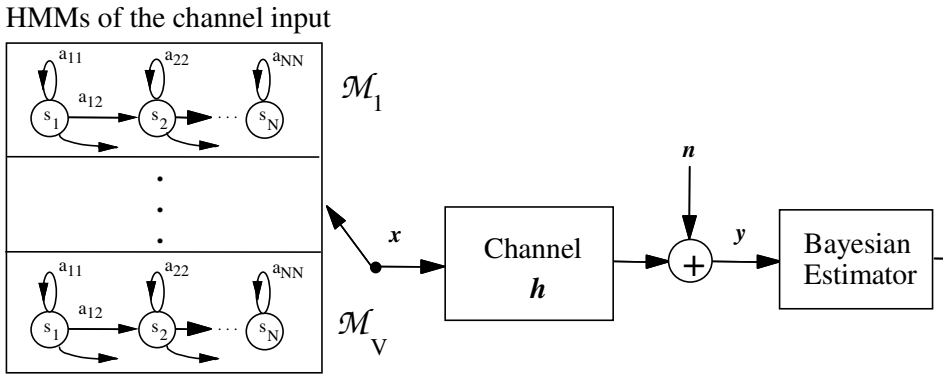


Figure 15.8 Illustration of a channel with the input modelled by a set of HMMs.

where $f_{\boldsymbol{X}|\mathcal{M},\mathbf{S}}(\boldsymbol{X} \mid \mathcal{M}_i, \mathbf{s})$ is the likelihood that the sequence $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(N-1)]$ was generated by the state sequence $\boldsymbol{s}=[s(0), ..., s(N-1)]$ of the model $\mathcal{M}_i$, and $P_{s|\mathcal{M}}(\boldsymbol{s} \mid \mathcal{M}_i)$ 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

$$a_{ij} = P\big(s(m) = j | s(m-1) = i\big) \tag{15.66}$$

where a_{ij} is the probability of making a transition from state i to state j. The HMM state observation probability is often modelled by a multivariate Gaussian pdf as

$$f_{\boldsymbol{X}|\mathcal{M},\mathbf{S}}(\boldsymbol{x} \mid \mathcal{M}_i, s) = \frac{1}{(2\pi)^{P/2} \mid \boldsymbol{\Sigma}_{\boldsymbol{xx},s} \mid^{1/2}} \exp\left\{-\frac{1}{2}[\boldsymbol{x} - \boldsymbol{\mu}_{x,s}]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s}^{-1} [\boldsymbol{x} - \boldsymbol{\mu}_{x,s}]\right\} \tag{15.67}$$

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 equalization problem can be stated as follows: Given a sequence of N P-dimensional channel output vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), ..., \boldsymbol{y}(N-1)]$, and the prior knowledge that the channel input

sequence is drawn from a set of V HMMs $\mathcal{M}=\{\mathcal{M}_i\ i=1, ..., V\}$, 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

$$f_{\mathcal{M},\boldsymbol{H}|\boldsymbol{Y}}(\mathcal{M}_i,\boldsymbol{h}\,|\,\boldsymbol{Y}) = P_{\mathcal{M}|\boldsymbol{H},\boldsymbol{Y}}(\mathcal{M}_i|\boldsymbol{h},\boldsymbol{Y})\, f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}\,|\,\boldsymbol{Y}) \tag{15.68}$$

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

$$f_{\boldsymbol{H}|\mathcal{M},\boldsymbol{Y}}(\boldsymbol{h}|\mathcal{M}_i,\boldsymbol{Y}) = \frac{1}{f_{\boldsymbol{Y}|\mathcal{M}}(\boldsymbol{Y}\,|\,\mathcal{M}_i)}\, f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y}|\mathcal{M}_i,\boldsymbol{h})\, f_{\boldsymbol{H}|\mathcal{M}}(\boldsymbol{h}\,|\,\mathcal{M}_i) \tag{15.69}$$

The likelihood of the observation sequence, given the channel and the input word model, can be expressed as

$$f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y}\,|\,\mathcal{M}_i,\boldsymbol{h}) = f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{Y}-\boldsymbol{h}\,|\,\mathcal{M}_i) \tag{15.70}$$

where it is assumed that the channel output is transformed into cepstral variables so that the channel distortion is additive. For a given input model $\mathcal{M}_i$, and state sequence $\boldsymbol{s}=[s(0), s(1), ..., s(N-1)]$, the pdf of a sequence of N independent observation vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), \boldsymbol{y}(1), ..., \boldsymbol{y}(N-1)]$ is

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{H},\mathbf{S},\mathcal{M}}(\boldsymbol{Y}\,|\,\boldsymbol{h},\mathbf{s},\mathcal{M}_i) &= \prod_{m=0}^{N-1} f_{\boldsymbol{X}|S,\mathrm{M}}(\boldsymbol{y}(m)-\boldsymbol{h}\,|\,s(m),\mathcal{M}_i) \\ &= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{\mathrm{P}/2}\,|\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}|^{1/2}} \exp\left\{-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]\right\} \end{aligned} \tag{15.71}$$

Taking the derivative of the log-likelihood of Equation (15.71) with respect to the channel vector $\boldsymbol{h}$ yields a maximum likelihood channel estimate as

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \sum_{m=0}^{N-1} \left(\sum_{k=0}^{N-1} \boldsymbol{\Sigma}_{xx,s(k)}^{-1} \right)^{-1} \boldsymbol{\Sigma}_{xx,s(m)}^{-1} \left(\boldsymbol{y}(m) - \boldsymbol{\mu}_{x,s(m)} \right) \quad (15.72)$$

Note that when all the state observation covariance matrices are identical the channel estimate becomes

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \frac{1}{N} \sum_{m=0}^{N-1} \left(\boldsymbol{y}(m) - \boldsymbol{\mu}_{\boldsymbol{x},s(m)} \right) \quad (15.73)$$

The ML estimate of Equation (15.73) 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.

15.4.5 MAP Channel Estimate Based on HMMs

The conditional pdf of a channel $\boldsymbol{h}$ averaged over all HMMs can be expressed as

$$f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h} \mid \boldsymbol{Y}) = \sum_{i=1}^{V} \sum_{\boldsymbol{s}} f_{\boldsymbol{H}|\boldsymbol{Y},\boldsymbol{S},\mathcal{M}}(\boldsymbol{h}|\boldsymbol{Y},\boldsymbol{s},\mathcal{M}_i) P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}_i) P_{\mathcal{M}}(\mathcal{M}_i) \quad (15.74)$$

where $P_{\mathcal{M}}(\mathcal{M}_i)$ is the prior pmf of the input words. Given a sequence of N P-dimensional observation vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), ..., \boldsymbol{y}(N-1)]$, the posterior pdf of the channel $\boldsymbol{h}$ along a state sequence $\boldsymbol{s}$ of an HMM $\mathcal{M}_i$ is defined as

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{H},\mathbf{S},\mathbf{M}}(\boldsymbol{h}|\boldsymbol{Y},\mathbf{s},\mathcal{M}_i) &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{Y})} f_{\boldsymbol{Y}|\boldsymbol{H},\mathbf{S},\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{h},\mathbf{s},\mathcal{M}_i) f_{\boldsymbol{H}}(\boldsymbol{h}) \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{Y})} \prod_{m=0}^{N-1} \frac{1}{(2\pi)^P |\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}|^{1/2} |\boldsymbol{\Sigma}_{\boldsymbol{hh}}|^{1/2}} \exp\left\{ -\frac{1}{2} \left[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{x},s(m)} \right]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1} \left[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{x},s(m)} \right] \right\} \\ &\quad \times \exp\left[-\frac{1}{2} (\boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{h}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1} (\boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{h}}) \right] \end{aligned} \quad (15.75)$$

where it is assumed that each state of the HMM has a Gaussian distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{x},s(m)}$ 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 (15.75), can be obtained as

$$\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y},\mathbf{s},\mathcal{M}_i)=\sum_{m=0}^{N-1}\left[\sum_{k=0}^{N-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{xx},s(k)}^{-1}+\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\right)\right]^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]$$
$$+\left[\sum_{k=0}^{N-1}\left(\boldsymbol{\Sigma}_{\boldsymbol{xx},s(k)}^{-1}+\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\right)\right]^{-1}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}\boldsymbol{\mu}_{\boldsymbol{h}} \tag{15.76}$$

The MAP estimate of the channel over all state sequences of all HMMs can be obtained as

$$\hat{\boldsymbol{h}}(\boldsymbol{Y})=\sum_{i=1}^{V}\sum_{\mathbf{S}}\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y},\mathbf{s},\mathcal{M}_i)P_{\mathbf{S}|\mathcal{M}}(\mathbf{s}\mid\mathcal{M}_i)P_{\mathcal{M}}(\mathcal{M}_i) \tag{15.77}$$

15.4.6 Implementations of HMM-Based Deconvolution

In this section, we consider three implementation methods for HMM-based channel equalization.

Method I: Use of the Statistical Averages Taken Over All HMMs

A simple approach to blind equalization, 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

$$\boldsymbol{\mu}_{\boldsymbol{x}}=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\boldsymbol{\mu}_{\mathcal{M}_i,j},\quad \boldsymbol{\Sigma}_{\boldsymbol{xx}}=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\boldsymbol{\Sigma}_{\mathcal{M}_i,j} \tag{15.78}$$

where $\boldsymbol{\mu}_{\mathcal{M}_i,j}$ and $\boldsymbol{\Sigma}_{\mathcal{M}_i,j}$ are the mean and the covariance of the j^{th} state of the i^{th} HMM, 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

$$\hat{\boldsymbol{h}}^{ML} = (\bar{\boldsymbol{y}} - \boldsymbol{\mu}_{\mathbf{x}}) \tag{15.79}$$

where $\bar{\boldsymbol{y}}$ is the time-averaged channel output. The estimate of the channel input is

$$\hat{\boldsymbol{x}}(m) = \boldsymbol{y}(m) - \hat{\boldsymbol{h}}^{ML} \tag{15.80}$$

Using the averages over all states and models, the MAP channel estimate becomes

$$\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y}) = \sum_{m=0}^{N-1} (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{hh} \ (\boldsymbol{y}(m) - \boldsymbol{\mu}_{x}) + (\boldsymbol{\Sigma}_{xx} + \boldsymbol{\Sigma}_{hh})^{-1} \boldsymbol{\Sigma}_{xx} \ \boldsymbol{\mu}_{h} \tag{15.81}$$

Method II: Hypothesised-Input HMM Equalization

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 15.9):

For i =1 to number of words V {

step 1 Using each HMM, $\mathcal{M}_i$, make an estimate of the channel, $\hat{\boldsymbol{h}}_i$,

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 $[\hat{\mathbf{x}}(m)]$. }

Select the channel estimate associated with the most probable word.

Figure 15.10 shows the ML channel estimates of two channels using unweighted average and hypothesised-input methods.

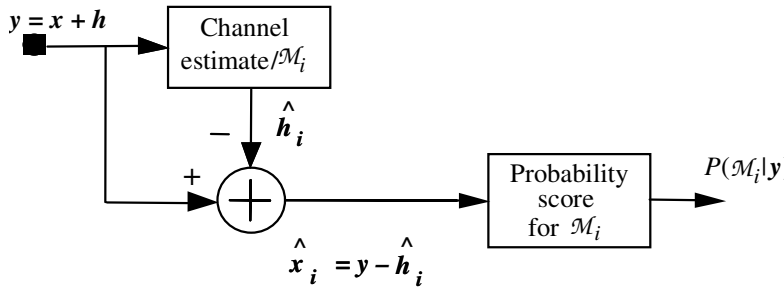


Figure 15.9 Hypothesised channel estimation procedure.

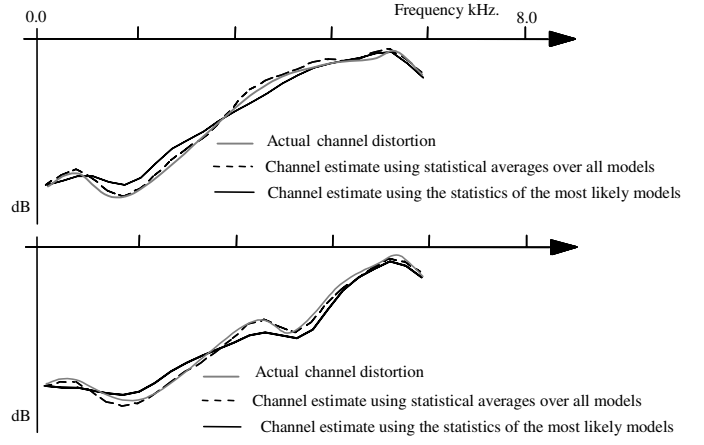


Figure 15.10 Illustration of actual and estimated channel response for two channels.

Method III: Decision-Directed Equalization

Blind adaptive equalizers are often composed of two distinct sections: an adaptive linear equalizer followed by a non-linear estimator to improve the equalizer 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 equalizer adaptation. The use of the output of the non-linear estimator as the desired signal assumes that the linear equalization 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 equalizer locks into, and cancels a large part of the channel distortion is to use a startup, equalizer training period during which a known signal is transmitted.

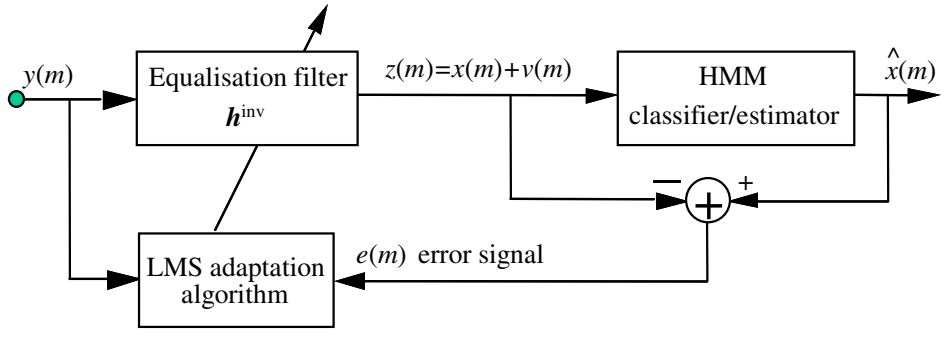


Figure 15.11 A decision-directed equalizer.

Figure 15.11 illustrates a blind equalizer 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 equalizer $z(m)$ is expressed as the sum of the input to the channel $x(m)$ and a so-called convolutional noise term $v(m)$ as

$$z(m)=x(m)+v(m) \tag{15.82}$$

The HMM may incorporate state-based Wiener filters for suppression of the convolutional noise $v(m)$ as described in Section 5.5. Assuming that the LMS adaptation method is employed, the adaptation of the equalizer coefficient vector is governed by the following recursive equation:

$$\hat{h}^{\text{inv}}(m)=\hat{h}^{\text{inv}}(m-1)+\mu e(m)y(m) \tag{15.83}$$

where $\hat{h}^{\text{inv}}(m)$ is an estimate of the optimal inverse channel filter, μ is an adaptation step size and the error signal $e(m)$ is defined as

$$e(m)=\hat{x}^{HMM}(m)-z(m) \tag{15.84}$$

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.

15.5 Blind Equalization for Digital Communication Channels

High speed transmission of digital data over analog channels, such as telephone lines or a radio channels, requires adaptive equalization 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 intersymbol interference (ISI), illustrated in Figure 15.12.

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 equalization.

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

$$x(t)=\sum_k a(k)r(t-kT_s) \tag{15.85}$$

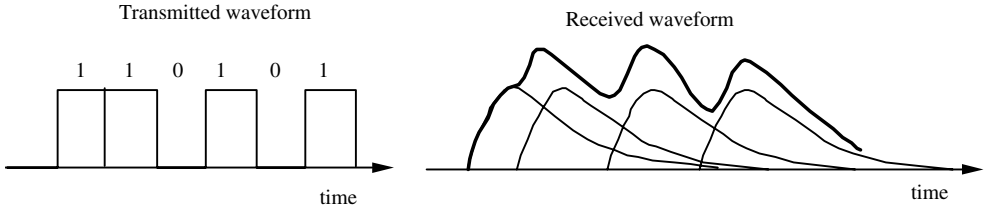


Figure 15.12 Illustration of intersymbol interference in a binary pulse amplitude modulation system.

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:

$$y(t)=\int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \tag{15.86}$$

where $h(t)$ is the channel impulse response. The sampled version of the channel output is given by the following discrete-time equation:

$$y(m)=\sum_k h_k\, x(m-k) \tag{15.87}$$

To remove the channel distortion, the sampled channel output $y(m)$ is passed to an equalizer with an impulse response $\hat{h}_k^{\text{inv}}$. The equalizer output $z(m)$ is given as

$$\begin{aligned} z(m) &= \sum_k \hat{h}_k^{\text{inv}}\, y(m-k) \\ &= \sum_j x(m-j)\sum_k \hat{h}_k^{inv}\, h_{j-k} \end{aligned} \tag{15.88}$$

where Equation (15.87) is used to obtain the second line of Equation (15.88). The ideal equalizer 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 equalizer. From Equation (15.88), the channel distortion would be cancelled if

$$h_m^c = h_m * \hat{h}_m^{\text{inv}} = \delta(m-D) \tag{15.89}$$

where h_m^c is the combined impulse response of the cascade of the channel and the equalizer. A particular form of channel equalizer, for the elimination of ISI, is the Nyquist *zero-forcing* filter, where the impulse response of the combined channel and equalizer is defined as

$$h^c(kT_s+D)=\begin{cases}1, & k=0\\ 0, & k\neq 0\end{cases} \tag{15.90}$$

Note that in Equation (15.90), at the sampling instants the channel distortion is cancelled, and hence there is no ISI at the sampling instants. A function that satisfies Equation (15.90) is the sinc function $h^c(t)=\sin(\pi f_s t)/\pi f_s t$, where $f_s=1/T_s$. Zero-forcing methods are sensitive to deviations of $h^c(t)$ from the requirement of Equation (15.90), and also to jitters in the synchronisation and the sampling process.

15.5.1 LMS Blind Equalization

In this section, we consider the more general form of the LMS-based adaptive equalizer 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 equalization, the desired signal (which is the channel input) is not available. The use of an adaptive filter for blind equalization, requires an internally generated desired signal as illustrated in Figure 15.13. Digital blind equalizers are composed of two distinct sections: an adaptive equalizer 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 equalizer adaptation. A method of ensuring that the equalizer removes a large part of the channel distortion is to use a start-up, equalizer training, period during which a known signal is transmitted.

Assuming that the LMS adaptation method is employed, the adaptation of the equalizer coefficient vector is governed by the following recursive equation:

$$\hat{\boldsymbol{h}}^{\text{inv}}(m)=\hat{\boldsymbol{h}}^{\text{inv}}(m-1)+\mu e(m)\boldsymbol{y}(m) \tag{15.91}$$

where $\hat{\boldsymbol{h}}^{\text{inv}}(m)$ is an estimate of the optimal inverse channel filter $\boldsymbol{h}^{\text{inv}}$, the scalar μ is the adaptation step size, and the error signal $e(m)$ is defined as

$$\begin{aligned} e(m) &= \psi(z(m)) - z(m) \\ &= \hat{x}(m) - z(m) \end{aligned} \tag{15.92}$$

where $\hat{x}(m) = \psi(z(m))$ is a non-linear estimate of the channel input. For example, in a binary communication system with an input alphabet $\{\pm a\}$ we

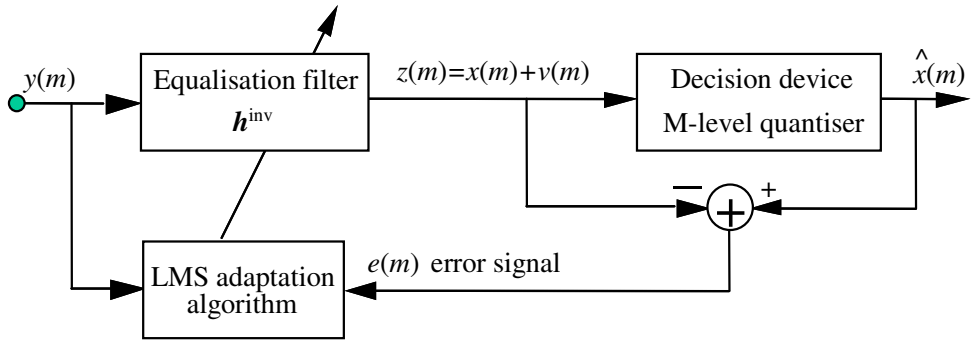


Figure 15.13 Configuration of an adaptive channel equalizer with an estimate of the channel input used as an "internally" generated desired signal

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 equalizer removes a large part of the channel distortion, the equalizer output can be expressed as the sum of the desired signal (the channel input) plus an uncorrelated additive noise term:

$$z(m) = x(m) + v(m) \tag{15.93}$$

where $v(m)$ is the so-called convolutional noise defined as

$$\begin{aligned} v(m) &= x(m) - \sum_k \hat{h}_k^{\text{inv}} y(m-k) \\ &= \sum_k (h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}) y(m-k) \end{aligned} \tag{15.94}$$

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 equalizer, each sample of convolutional noise is affected by many samples of the input process. From the central limit theorem, the convolutional noise $e(m)$ can be modelled by a zero-mean Gaussian process as

$$f_E(e(m)) = \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{e^2(m)}{2\sigma_e^2}\right) \tag{15.95}$$

where σ_e^2, the noise variance, can be estimated using the recursive time-update equation

$$\sigma_e^2(m) = \rho\sigma_e^2(m-1) + (1-\rho)e^2(m) \tag{15.96}$$

where $\rho < 1$ is the adaptation factor. The Bayesian estimate of the channel input given the equalizer output can be expressed in a general form as

$$\hat{x}(m) = \arg\min_{\hat{x}(m)} \int_X C(x(m), \hat{x}(m)) f_{X|Z}(x(m) \mid z(m))\, dx(m) \tag{15.97}$$

where $C(x(m), \hat{x}(m))$ is a cost function and $f_{X|Z}(x(m)|z(m))$ is the posterior pdf of the channel input signal. The choice of the cost function determines the type of the estimator as described in Chapter 4. Using a uniform cost function in Equation (15.97) yields the maximum a posteriori (MAP) estimate

$$\begin{aligned} \hat{x}^{MAP}(m) &= \arg\max_{x(m)} f_{X|Z}(x(m) \mid z(m)) \\ &= \arg\max_{x(m)} f_E(z(m) - x(m)) P_X(x(m)) \end{aligned} \tag{15.98}$$

Now, as an example consider an M-ary pulse amplitude modulation system, and let $\{a_i \quad i=1, ..., M\}$ denote the set of M pulse amplitudes with a probability mass function

$$P_X(x(m)) = \sum_{i=1}^{M} P_i \delta(x(m) - a_i) \tag{15.99}$$

The pdf of the equalizer output $z(m)$ can be expressed as the mixture pdf

$$f_Z(z(m)) = \sum_{i=1}^{M} P_i f_E(x(m) - a_i) \tag{15.100}$$

The posterior density of the channel input is

$$P_{X|Z}(x(m)=a_i|z(m))=\frac{1}{f_Z(z(m))}f_E(z(m)-a_i)P_X(x(m)=a_i) \quad (15.101)$$

and the MAP estimate is obtained from

$$\hat{x}^{MAP}(m)=\arg\max_{a_i}(f_E(z(m)-a_i)P_X(x(m)=a_i)) \quad (15.102)$$

Note that the classification of the continuous-valued equalizer 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 (102) yields

$$\hat{x}^{MAP}(m)=\arg\max_{a_i}\left[P_X(x(m)=a_i)\exp\left\{-\frac{[z(m)-a_i]^2}{2\sigma_e^2}\right\}\right] \quad (15.103)$$

Note that when the symbols are equiprobable, the MAP estimate reduces to a simple threshold decision device. Figure 15.13 shows a channel equalizer followed by an M-level quantiser. In this system, the output of the equalizer 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 equalizer adaptation.

15.5.2 Equalization of a Binary Digital Channel

Consider a binary PAM communication system with an input symbol alphabet $\{a_0, a_1\}$ and symbol probabilities $P(a_0)=P_0$ and $P(a_1)=P_1=1-P_0$. The pmf of the amplitude of the channel input signal can be expressed as

$$P(x(m))=P_0\delta(x(m)-a_0)+P_1\delta(x(m)-a_1) \quad (15.104)$$

Assume that at the output of the linear adaptive equalizer in Figure 15.13, the convolutional noise $v(m)$ is a zero-mean Gaussian process with variance

σ_v^2. Therefore the pdf of the equalizer output $z(m)=x(m)+v(m)$ is a mixture of two Gaussian pdfs and can be described as

$$f_Z(z(m))=\frac{P_0}{\sqrt{2\pi}\sigma_v}\exp\left\{-\frac{[z(m)-a_0]^2}{2\sigma_v^2}\right\}+\frac{P_1}{\sqrt{2\pi}\sigma_v}\exp\left\{-\frac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \tag{15.105}$$

The MAP estimate of the channel input signal is

$$\hat{x}(m)=\begin{cases} a_0 & \text{if } \frac{P_0}{\sqrt{2\pi}\sigma_v}\exp\left\{-\frac{[z(m)-a_0]^2}{2\sigma_v^2}\right\} < \frac{P_1}{\sqrt{2\pi}\sigma_v}\exp\left\{-\frac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \\ a_1 & \text{otherwise} \end{cases} \tag{15.106}$$

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

$$e(m) = z(m) - sgn(z(m))a \tag{15.107}$$

Figure 15.14 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,

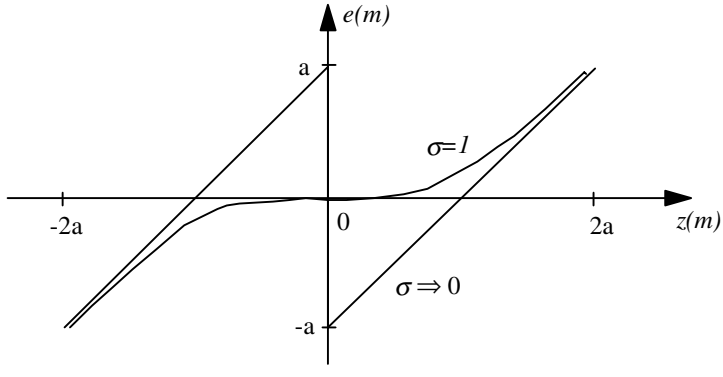


Figure 15.14 Comparison of the error functions produced by the hard non-linearity of a sign function Equation (15.107) and the soft non-linearity of Equation

and a decision based on the sign of $z(m)$ is most likely to be incorrect.

A large error signal based on an incorrect decision would have an unsettling effect on the convergence of the adaptive equalizer. It is desirable to have an error function that produces small error signals when $z(m)$ is around zero. Nowlan and Hinton proposed a soft non-linearity of the following form

$$e(m)=z(m)-\frac{e^{2az(m)/\sigma^2}-1}{e^{2az(m)/\sigma^2}+1}a \tag{15.108}$$

The error $e(m)$ is small when the magnitude of $z(m)$ is small and large when magnitude of $z(m)$ is large.

15.6 Equalization Based on Higher-Order Statistics

The second-order statistics of a random process, namely the autocorrelation or its Fourier transform the power spectrum, are central to the development the linear estimation theory, and form the basis of most statistical signal processing methods such as Wiener filters and linear predictive models. An attraction of the correlation function is that a Gaussian process, of a known mean vector, can be completely described in terms of the covariance matrix, and many random processes can be well characterised by Gaussian or mixture Gaussian models. A shortcoming of second-order statistics is that they do not include the phase characteristics of the process. Therefore, given the channel output, it is not possible to estimate the channel phase from the second-order statistics. Furthermore, as a Gaussian process of known mean depends entirely on the autocovariance function, it follows that blind deconvolution, based on a Gaussian model of the channel input, cannot estimate the channel phase.

Higher-order statistics, and the probability models based on them, can model both the magnitude and the phase characteristics of a random process. In this section, we consider blind deconvolution based on higher-order statistics and their Fourier transforms known as the higher-order spectra. The prime motivation in using 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.

15.6.1 Higher-Order Moments, Cumulants and Spectra

The k^{th} *order* moment of a random variable X is defined as

$$m_k = \mathcal{E}[x^k] = (-j)^k \left. \frac{\partial^k \Phi_X(\omega)}{\partial \omega^k} \right|_{\omega=0} \tag{15.109}$$

where $\Phi_X(\omega)$ is the *characteristic function* of the random variable X defined as

$$\Phi_X(\omega) = \mathcal{E}[\exp(j\omega x)] \tag{15.110}$$

From Equations (15.109) and (15.110), the first moment of X is $m_1 = \mathcal{E}[x]$, the second moment of X is $m_2 = \mathcal{E}[x^2]$, and so on. The joint k^{th} order moment ($k = k_1 + k_2$) of two random variables X_1 and X_2 is defined as

$$\mathcal{E}[x_1^{k_1} x_2^{k_2}] = (-j)^{k_1+k_2} \left. \frac{\partial^{k_1} \partial^{k_2} \Phi_{X_1 X_2}(\omega_1, \omega_2)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2}} \right|_{\omega_1=\omega_2=0} \tag{15.111}$$

and in general the joint k^{th} order moment of N random variables is defined as

$$m_k = \mathcal{E}[x_1^{k_1} x_2^{k_2} \ldots x_N{}^{k_N}] = (-\mathrm{j})^k \left. \frac{\partial^k \Phi(\omega_1, \omega_2, \ldots, \omega_N)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \cdots \partial \omega_N^{k_N}} \right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \tag{15.112}$$

where $k = k_1 + k_2 + \ldots + k_N$ and the joint characteristic function is

$$\Phi(\omega_1, \omega_2, \ldots, \omega_N) = \mathcal{E}[\exp(j\omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_N x_N)] \tag{15.113}$$

Now the higher-order moments can be applied for characterization of discrete-time random processes. The k^{th} order moment of a random process $x(m)$ is defined as

$$m_x(\tau_1, \tau_2, \ldots, \tau_{K-1}) = \mathcal{E}[x(m), x(m+\tau_1) x(m+\tau_2) \cdots x(m+\tau_{k-1})] \tag{15.114}$$

Note that the second-order moment $\mathcal{E}[x(m)x(m+\tau)]$ is the autocorrelation function.

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

$$C_X(\omega)=\ln\Phi_X(\omega)=\ln\mathcal{E}[\exp(\mathrm{j}\omega x)] \tag{15.115}$$

Using a Taylor series expansion of the term $\mathcal{E}[\exp(j\omega x)]$ in Equation (15.115) the cumulant generating function can be expanded as

$$C_X(\omega)=\ln\left(1+m_1(\mathrm{j}\omega)+\frac{m_2}{2!}(\mathrm{j}\omega)^2+\frac{m_3}{3!}(\mathrm{j}\omega)^3+\cdots+\frac{m_n}{n!}(\mathrm{j}\omega)^n+\cdots\right) \tag{15.116}$$

where $m_k=\mathcal{E}[x^k]$ is the k^{th} moment of the random variable x. The k^{th} order cumulant of a random variable is defined as

$$c_k=(-\mathrm{j})^k\left.\frac{\partial^k C_X(\omega)}{\partial\omega^k}\right|_{\omega=0} \tag{15.117}$$

From Equations (15.116) and (15.117), we have

$$c_1=m_1 \tag{15.118}$$

$$c_2=m_2-m_1^2 \tag{15.119}$$

$$c_3=m_3-3m_1m_2+2m_1^2 \tag{15.120}$$

and so on. The general form of the k^{th} order ($k=k_1+k_2+\cdots+k_N$) joint cumulant generating function is

$$c_{k_1\cdots k_N} = (-\mathrm{j})^{k_1+\cdots+k_N} \left. \frac{\partial^{k_1+\cdots+k_N} \ln\Phi_X(\omega_1,\cdots,\omega_N)}{\partial\omega_1^{k_1}\cdots\partial\omega_N^{k_N}} \right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \tag{15.121}$$

The cumulants of a zero mean random process $x(m)$ are given as

$$c_x = \mathcal{E}[x(k)] = m_x = 0 \qquad \text{(mean)} \tag{15.122}$$

$$\begin{aligned} c_x(k) &= \mathcal{E}[x(m)x(m+k)] - \mathcal{E}[x(m)]^2 \\ &= m_x(k) - m_x^2 = m_x(k) \end{aligned} \qquad \text{(covariance)} \tag{15.123}$$

$$\begin{aligned} c_x(k_1,k_2) &= m_x(k_1,k_2) - m_x[m_x(k_1) + m_x(k_2) + m_x(k_2-k_1)] + 2(m_x)^3 \\ &= m_x(k_1,k_2) \end{aligned} \qquad \text{(skewness)} \tag{15.124}$$

$$\begin{aligned} c_x(k_1,k_2,k_3) = m_x(k_1,k_2,k_3) &- m_x(k_1)m_x(k_3-k_2) \\ &- m_x(k_2)m_x(k_3-k_1) - m_x(k_3)m_x(k_2-k_1) \end{aligned} \tag{15.125}$$

and so on. Note that $m_x(k_1, k_2, ..., k_N) = \mathcal{E}[x(m)x(m+k_1), x(m+k_2), ..., x(m+k_N)]$. The general formulation of the k^{th} order cumulant of a random process $x(m)$ (Rosenblatt) is defined as

$$c_x(k_1,k_2,\ldots,k_n) = m_x(k_1,k_2,\ldots,k_n) - m_x^G(k_1,k_2,\ldots,k_n) \qquad \textit{for} \quad n = 3, 4, \ldots \tag{15.126}$$

where $m_x^G(k_1,k_2,\ldots,k_n)$ is the k^{th} order moment of a Gaussian process having the same mean and autocorrelation as the random process $x(m)$. From Equation (15.126), it follows that for a Gaussian process, the cumulants of order greater than 2 are identically zero.

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

$$C_X(\omega_1,\ldots,\omega_{k-1})=\frac{1}{(2\pi)^{k-1}}\sum_{\tau_1=-\infty}^{\infty}\cdots\sum_{\tau_{k-1}=-\infty}^{\infty}c_x(\tau_1,\ldots,\tau_{k-1})e^{-j(\omega_1\tau_1+\cdots+\omega_{k-1}\tau_{k-1})} \tag{15.127}$$

For the case k=2, the second-order spectrum is the power spectrum given as

$$C_X(\omega)=\frac{1}{2\pi}\sum_{\tau=-\infty}^{\infty}c_x(\tau)e^{-j\omega\tau} \tag{15.128}$$

The *bi-spectrum* is defined as

$$C_X(\omega_1,\omega_2)=\frac{1}{(2\pi)^2}\sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty}c_x(\tau_1,\tau_2)e^{-j(\omega_1\tau_1+\omega_2\tau_2)} \tag{15.129}$$

and the *tri-spectrum* is

$$C_X(\omega_1,\omega_2,\omega_3)=\frac{1}{(2\pi)^3}\sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty}\sum_{\tau_3=-\infty}^{\infty}c_x(\tau_1,\tau_2,\tau_3)e^{-j(\omega_1\tau_1+\omega_2\tau_2+\omega_3\tau_3)} \tag{15.130}$$

Since the term $e^{j\omega t}$ is periodic with a period of 2π, it follows that higher order spectra are periodic in each ω_k with a period of 2π.

15.6.2 Higher-Order Spectra of Linear Time-Invariant Systems

Consider a linear time-invariant system with an impulse response sequence $\{h_k\}$, input signal $x(m)$ and output signal $y(m)$. The relation between the k^{th}-order cumulant spectra of the input and output signals is given by

$$C_Y(\omega_1,\ldots,\omega_{k-1})=H(\omega_1)\cdots H(\omega_{k-1})H^*(\omega_1+\cdots+\omega_{k-1})C_X(\omega_1,\ldots,\omega_{k-1}) \tag{15.131}$$

where $H(\omega)$ is the frequency response of the linear system $\{h_k\}$. The magnitude of the k^{th}-order spectrum of the output signal is given as

$$|C_Y(\omega_1,\ldots,\omega_{k-1})|=|H(\omega_1)|\cdots|H(\omega_{k-1})|\,|H(\omega_1+\cdots+\omega_{k-1})|\,|C_X(\omega_1,\ldots,\omega_{k-1})| \tag{15.132}$$

and the phase of the k^{th}-order spectrum is

$$\Phi_Y(\omega_1,\ldots,\omega_{k-1})=\Phi_H(\omega_1)+\cdots+\Phi_H(\omega_{k-1})-\Phi_H(\omega_1+\cdots+\omega_{k-1})+\Phi_X(\omega_1,\ldots,\omega_{k-1}) \tag{15.133}$$

15.6.3 Blind Equalization Based on Higher-Order Cepstra

In this section, we consider blind equalization 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

$$H(z)=GH_{\min}(z)H_{\max}(z^{-1})z^{-D} \tag{15.134}$$

$$H_{\min}(z)=\prod_{i=1}^{P_1}(1-\alpha_i z^{-1}),\quad |\alpha_i|<1 \tag{15.135}$$

$$H_{\max}(z^{-1})=\prod_{i=1}^{P_2}(1-\beta_i z),\quad |\beta_i|<1 \tag{15.136}$$

where G is a gain factor, $H_{\min}(z)$ is a minimum-phase polynomial with all its zeros inside the unit circle, $H_{\max}(z^{-1})$ is a maximum-phase polynomial with all its zeros outside the unit circle, and z^{-D} inserts D unit delays in order to make Equation (15.134) causal. The complex cepstrum of $H(z)$ is defined as

$$h_c(m)=Z^{-1}(\ln H(z)) \tag{15.137}$$

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

$$P_Y(\omega)=\frac{\sigma_x^2}{2\pi}H(\omega)H^*(\omega) \tag{15.138}$$

The cepstrum of the power spectrum of $y(m)$ is defined as

$$\begin{aligned} y_c(m) &= IDFT(\ln P_Y(\omega)) \\ &= IDFT\left(\ln\left(\sigma_x^2 G^2/2\pi\right)+\ln H_{\min}(\omega)+H_{\max}(-\omega)+\ln H_{\min}^*(\omega)+H_{\max}^*(-\omega)\right) \end{aligned} \tag{15.139}$$

where IDFT is the inverse discrete Fourier transform. Substituting Equations (15.135) and (15.36) in (15.139), the cepstrum can be expressed as

$$y_c(m)=\begin{cases} \ln\left(G^2\sigma_x^2/2\pi\right), & m=0 \\ -\left(A^{(m)}+B^{(m)}\right)/m, & m>0 \\ \left(A^{(-m)}+B^{(-m)}\right)/m, & m<0 \end{cases} \tag{15.140}$$

where $A^{(m)}$ and $B^{(m)}$ are defined as

$$A^{(m)}=\sum_{i=1}^{P_1}\alpha_i^m \tag{15.141}$$

$$B^{(m)}=\sum_{i=1}^{P_2}\beta_i^m \tag{15.142}$$

Note from Equation (15.140) that the along the index m, the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ overlap and cannot be separated.

Bi-Cepstrum

The bi-cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the bi-spectrum:

$$y_c(m_1,m_2)=IDFT_2[\log C_Y(\omega_1,\omega_2)] \tag{15.143}$$

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

$$C_Y(\omega_1,\omega_2)=H(\omega_1)H(\omega_2)H^*(\omega_1+\omega_2)C_X(\omega_1,\omega_2) \tag{15.144}$$

Assuming that the input $x(m)$ of the linear time-invariant system $\{h_k\}$ is an uncorrelated non-Gaussian process, the bi-spectrum of the output can be written as

$$\begin{aligned} C_Y(\omega_1,\omega_2)=&\frac{\gamma_x^{(3)}G^3}{(2\pi)^2}H_{\min}(\omega_1)H_{\max}(-\omega_1)H_{\min}(\omega_2)H_{\max}(-\omega_2) \\ &\times H_{\min}^*(\omega_1+\omega_2)H_{\max}^*(-\omega_1-\omega_2) \end{aligned} \tag{15.145}$$

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 (15.145) yields

$$\begin{aligned} \ln C_y(\omega_1,\omega_2)=&\ln|A|+\ln H_{\min}(\omega_1)+\ln H_{\max}(-\omega_1)+\ln H_{\min}(\omega_2)+\ln H_{\max}(-\omega_2) \\ &\ln H_{\min}^*(\omega_1+\omega_2)+\ln H_{\max}^*(-\omega_1-\omega_2) \end{aligned} \tag{15.146}$$

where $A=\gamma_x^{(3)}G^3/(2\pi)^2$. The bi-cepstrum is obtained through the inverse Discrete Fourier transform of Equation (15.146) as

$$y_c(m_1,m_2)=\begin{cases} \ln|A|, & m_1=m_2=0 \\ -A^{(m_1)}/m_1, & m_1>0, m_2=0 \\ -A^{(m_2)}/m_2, & m_2>0, m_1=0 \\ -B^{(-m_1)}/m_1, & m_1<0, m_2=0 \\ B^{(-m_2)}/m_2, & m_2<0, m_1=0 \\ -B^{(m_2)}/m_2, & m_1=m_2>0 \\ A^{(-m_2)}/m_2, & m_1=m_2<0 \\ 0, & \text{otherwise} \end{cases} \tag{15.147}$$

Note from Equation (15.147) that the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ are separated and appear in different regions of the bi-cepstrum indices m_1 and m_2.

The higher-order cepstral coefficients can be obtained either from the IDFT of higher-order spectra as in Equation (15.147) or using parametric methods as follows. In general, the cepstral and cumulant coefficients can be related by a convolutional equation. Pan and Nikias (1988) have shown that the recursive relation between the bi-cepstrum coefficients and the third-order cumulants of a random process is

$$y_c(m_1,m_2)*\left[-m_1 c_y(m_1,m_2)\right]=-m_1 c_y(m_1,m_2) \tag{15.148}$$

Substituting Equation (15.147) in Equation (15.148) yields

$$\sum_{i=1}^{\infty} A^{(i)}[c_x(m_1-i,m_2)-c_x(m_1+i,m_2+i)]+B^{(i)}[c_x(m_1-i,m_2-i)-c_x(m_1+i,m_2)] =-m_1 c_x(m_1,m_2) \tag{15.149}$$

The truncation of the infinite summation in Equation (15.149) provides an approximate equation as

$$\sum_{i=1}^{P} A^{(i)}[c_x(m_1-i,m_2)-c_x(m_1+i,m_2+i)] + \sum_{i=1}^{Q} B^{(i)}[c_x(m_1-i,m_2-i)-c_x(m_1+i,m_2)]\approx -m_1 c_x(m_1,m_2) \tag{15.150}$$

Equation (15.150) can be used to solve for the cepstral parameters $A^{(m)}$ and $B^{(m)}$.

Tri-Cepstrum

The tri-cepstrum of a signal $y(m)$ is defined as the inverse Fourier transform of the tri-spectrum:

$$y_c(m_1,m_2,m_3)=IDFT_3[\ln C_Y(\omega_1,\omega_2,\omega_3)] \tag{15.151}$$

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

$$C_Y(\omega_1,\omega_2,\omega_3)=H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3)C_X(\omega_1,\omega_2,\omega_3) \tag{15.152}$$

Assuming that the channel input $x(m)$ is uncorrelated, Equation (15.152) becomes

$$C_Y(\omega_1,\omega_2,\omega_3)=\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3) \tag{15.153}$$

where $\gamma_x^{(4)}/(2\pi)^3$ is the fourth-order cumulant of the input signal. Taking the logarithm of the tri-spectrum gives

$$\begin{aligned}\ln C_Y(\omega_1,\omega_2,\omega_3)=&\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}+\ln H_{\min}(\omega_1)+\ln H_{\max}(-\omega_1)+\ln H_{\min}(\omega_2)+\ln H_{\max}(-\omega_2)\\&+\ln H_{\min}(\omega_3)+\ln H_{\max}(-\omega_3)+\ln H^*_{\min}(\omega_1+\omega_2+\omega_3)+\ln H^*_{\max}(-\omega_1-\omega_2-\omega_3)\end{aligned} \tag{15.154}$$

From Equations (15.151) and (15.154), we have

$$y_c(m_1,m_2,m_3)=\begin{cases}\ln A, & m_1=m_2=m_3=0\\ -A^{(m_1)}/m_1, & m_1>0, m_2=m_3=0\\ -A^{(m_2)}/m_2, & m_2>0, m_1=m_3=0\\ -A^{(m_3)}/m_3, & m_3>0, m_1=m_2=0\\ B^{(-m_1)}/m_1, & m_1<0, m_2=m_3=0\\ B^{(-m_2)}/m_2, & m_2<0, m_1=m_3=0\\ B^{(-m_3)}/m_3, & m_3<0, m_1=m_2=0\\ -B^{(m_2)}/m_2, & m_1=m_2=m_3>0\\ A^{(m_2)}/m_2, & m_1=m_2=m_3<0\\ 0 & \text{otherwise}\end{cases} \tag{15.155}$$

where $A=\gamma_x^{(4)}G^4/(2\pi)^3$. Note from Equation (15.155) that the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ are separated and appear in different regions of the tri-cepstrum indices m_1, m_2 and m_3.

Calculation of Equalizer Coefficients from the Tri-cepstrum

Assuming that the channel z-transfer function can be described by Equation (15.134), the inverse channel can be written as

$$H^{\text{inv}}(z)=\frac{1}{H(z)}=\frac{1}{H_{\min}(z)H_{\max}(z^{-1})}=H_{\min}^{\text{inv}}(z)H_{\max}^{\text{inv}}(z^{-1}) \qquad (15.156)$$

where it is assumed that the channel gain G is unity. In the time domain Equation (15.156) becomes

$$h^{\text{inv}}(m)=h_{\min}^{\text{inv}}(m)*h_{\max}^{\text{inv}}(m) \qquad (15.157)$$

Pan and Nikias (1988) describe an iterative algorithm for estimation of the truncated impulse response of the maximum-phase and the minimum-phase factors of the inverse channel transfer function. Let $\hat{h}_{\min}^{\text{inv}}(i,m)$, $\hat{h}_{\max}^{\text{inv}}(i,m)$ denote the estimates of the m^{th} coefficients of the maximum-phase and minimum-phase parts of the inverse channel at the i^{th} iteration . The Pan and Nikias algorithm is the following:

(a) Initialisation

$$\hat{h}_{\min}^{\text{inv}}(i,0)=\hat{h}_{\max}^{\text{inv}}(i,0)=1 \qquad (15.158)$$

(b) Calculation of the minimum-phase polynomial

$$\hat{h}_{\min}^{\text{inv}}(i,m)=\frac{1}{m}\sum_{k=2}^{m+1}\hat{A}^{(k-1)}\hat{h}_{\min}^{\text{inv}}(i,m-k+1) \qquad i=1,\ ...,\ P_1 \qquad (15.159)$$

(c) Calculation of the maximum-phase polynomial

$$\hat{h}_{\max}^{\text{inv}}(i,m)=\frac{1}{m}\sum_{k=m+1}^{0}\hat{B}^{(1-k)}\hat{h}_{\max}^{\text{inv}}(i,m-k+1) \qquad i=-1,\ ...,\ -P_2 \qquad (15.160)$$

The maximum-phase and minimum-phase components of the inverse channel response are combined in Equation (15.157) to give the inverse channel equalizer.

15.7 Summary

In this chapter, we considered a number of different approaches to channel equalization. The chapter began with an introduction to models for channel distortions, the definition of an ideal channel equalizer, and the problems that arise in channel equalization 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 15.1, we considered the least square error Wiener equalizer. The Wiener equalizer 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 equalization method. Blind equalization is feasible only if some statistics of the channel input signal are available. In Section 15.2, we considered blind equalization using the power spectrum of the input signal. This method was introduced by Stockham for restoration of the magnitude spectrum of distorted acoustic recordings. In Section 15.3, we considered a blind deconvolution method based on the factorisation of a linear predictive model of the convolved signals.

Bayesian inference provides a framework for inclusion of the statistics of the channel input and perhaps also those of the channel environment. In Section 15.4, we considered Bayesian equalization methods, and studied the case where the channel input is modelled by a set of hidden Markov models. Section 15.5 introduced channel equalization methods for removal of intersymbol interference in digital telecommunication systems, and finally in Section 15.6, we considered the use of higher-order spectra for equalization 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**, pp. 385–399.

BELLINI S. (1986) Bussgang Techniques for Blind Equalization. IEEE GLOBECOM Conf. Rec., pp. 1634–1640.

BELLINI S. and ROCCA F. (1988) Near Optimal Blind Deconvolution. IEEE Proc. Int. Conf. Acoustics, Speech, and Signal Processing. ICASSP-88, pp. 2236–2239.

BELFIORE C.A. and PARK J.H. (1979) Decision Feedback Equalization. Proc. IEEE, 67, pp. 1143–1156.

GERSHO A. (1969) Adaptive Equalization of Highly Dispersive Channels for Data Transmission. Bell System Technical Journal, **48**, pp. 55–70.

GODARD D.N. (1974) Channel Equalization using a Kallman Filter for Fast Data Transmission. IBM J. Res. Dev., **18**, pp. 267–273.

GODARD D.N. (1980) Self-recovering Equalization and Carrier Tracking in a Two-Dimensional Data Communication System. IEEE Trans. Comm., **COM-28**, pp. 1867-75.

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, pp. 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**, pp. 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, pp. 85–88.

LUCKY R.W. (1965) Automatic Equalization of Digital Communications. Bell System technical Journal, **44**, pp. 547–588.

LUCKY R.W. (1965) Techniques for Adaptive Equalization of Digital Communication Systems. Bell System Technical Journal, **45**, pp. 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**, pp. 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 Technoplogy. EuroSpeech-93, **2**, pp. 1247-1250.

MONSEN P. (1971) Feedback Equalization for Fading Dispersive Channels. IEEE Trans. Information Theory, **IT-17**, pp. 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**, pp. 1911–1913.

NOWLAN S.J. and HINTON G.E. (1993) A Soft Decision-Directed Algorithm for Blind Equalization IEEE Transactions on Communications., **41**, No. 2, pp. 275–279.

PAN R. and NIKIAS C.L. (1988) Complex Cepstrum of Higher Order Cumulants and Non-minimum Phase Identification. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-36**, pp. 186–205.

PICCHI G. and PRATI G. (1987) Blind Equalization and Carrier Recovery using a Stop-and-Go Decision-Directd Algorithm, IEEE Trans. Commun, **COM-35**, pp. 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**, pp. 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., CANNON T.M. and INGEBRETSEN R.B (1975) Blind Deconvolution Through Digital Signal Processing. IEEE Proc., **63**, **4**, pp. 678-92.

QURESHI S.U. (1985) Adaptive Equalization. IEEE Proc. Vol 73, No. 9, pp. 1349–1387.

UNGERBOECK G. (1972) Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication. IBM J. Res. Dev., **16**, pp. 546-555.

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

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z