

# Digital Audio Restoration - a statistical model based approach

Simon J. Godsill and Peter J.W. Rayner

September 21, 1998



We dedicate this book to our families,  
especially to Rachel<sup>(+Rufus)</sup> and Ann.



## Preface

---

The application of digital signal processing (DSP) to problems in audio has been an area of growing importance since the pioneering DSP work of the 1960s and 70s. In the 1980s, DSP micro-chips became sufficiently powerful to handle the complex processing operations required for sound restoration in real-time, or close to real-time. This led to the first commercially available restoration systems, with companies such as CEDAR Audio Ltd. in the UK and Sonic Solutions in the US selling dedicated systems world-wide to recording studios, broadcasting companies, media archives and film studios. Vast amounts of important audio material, ranging from historic recordings of the last century to relatively recent recordings on analogue or even digital tape media, were noise-reduced and re-released on CD for the increasingly quality-conscious music enthusiast. Indeed, the first restorations were a revelation in that clicks, crackles and hiss could for the first time be almost completely eliminated from recordings which might otherwise be un-releasable in CD format.

Until recently, however, digital audio processing has required high-powered computational engines which were only available to large institutions who could afford to use the sophisticated digital remastering technology. With the advent of compact disc and other digital audio formats, followed by the increased accessibility of home computing, digital audio processing is now available to anyone who owns a PC with sound card, and will be of increasing importance, in association with digital video, as the multimedia revolution continues into the next millennium. Digital audio restoration will thus find increasing application to sound recordings from the internet, home recordings and speech, and high-quality noise-reducers will become a standard part of any computer system and hifi system, alongside speech recognisers and image processors.

In this book we draw upon extensive experience in the commercial world of sound restoration<sup>1</sup> and in the academic community, to give a comprehensive overview of the principles behind the current technology, as implemented in the commercial restoration systems of today. Furthermore, if the current technology can be regarded as a ‘first phase’ in audio restoration, then the later chapters of the book outline a ‘second phase’ of more sophisticated statistical methods which are aimed at achieving higher fidelity to the original recorded sound and at addressing problems which cannot currently be handled by commercial systems. It is anticipated that new methods such as these will form the basis of future restoration systems.

---

<sup>1</sup>Both authors were founding members of CEDAR (Computer Enhanced Digital Audio Restoration), the Cambridge-based audio restoration company.

We acknowledge with gratitude the help of numerous friends and colleagues in the production and research for this book: to Dr Bill Fitzgerald for advice and discussion on Bayesian methods; to Dr Anil Kokaram for support, friendship, humour and technical assistance throughout the research period; to Dr Saeed Vaseghi, whose doctoral research work initiated audio restoration research in Cambridge; to Dr Christopher Roads, whose unbounded energy and enthusiasm made the CEDAR project a reality; to Dave Betts, Gordon Reid and all at CEDAR; to an host of proof-readers who gave their assistance willingly and often at short notice, including Tim Clapp, Colin Campbell, Paul Walmsley, Dr Mike Davies, Dr Arnaud Doucet, Jacek Noga, Steve Armstrong and Rachel Godsill; and finally to the many others who have assisted directly and indirectly over the years.

Simon Godsill  
Cambridge, 1998.

## Glossary

---

AR	autoregressive
ARMA	autoregressive moving-average
CDF	cumulative distribution function
DFT	discrete Fourier transform
DTFT	discrete time Fourier transform
EM	expectation-maximisation
FFT	fast Fourier transform
FIR	finite impulse response
i.i.d.	independent, identically distributed
IIR	infinite impulse response
LS	least squares
MA	moving average
MAP	maximum <i>a posteriori</i>
MCMC	Markov chain Monte Carlo
ML	maximum likelihood
MMSE	minimum mean squared error
MSE	mean squared error
PDF	probability density function
PMF	probability mass function
RLS	recursive least squares
SNR	signal to noise ratio
w.r.t.	with respect to

---





## Notation

---

Obscure mathematical notation is avoided wherever possible. However, the following glossary of basic notation, which is adopted unless otherwise stated, may be a useful reference.

Scalars	Lower/upper case, e.g. $x_t$ or $E$
Column vectors	Bold lower case, e.g. $\mathbf{x}$
Matrices	Bold upper case, e.g. $\mathbf{M}$
$p(\cdot)$	Probability distribution (density or mass function)
$f(\cdot)$	Probability density function
$F(\cdot)$	Cumulative distribution function
$E[\cdot]$	Expected value
$N(\mu, \sigma^2)$ or $N(\theta \mu, \sigma^2)$	Univariate normal distribution, mean $\mu$ , covariance $\sigma^2$
$N_q(\boldsymbol{\mu}, \mathbf{C})$ or $N_q(\boldsymbol{\theta} \boldsymbol{\mu}, \mathbf{C})$	$q$ -variate normal distribution
$G(\alpha, \beta)$ or $G(v \alpha, \beta)$	Gamma distribution
$IG(\alpha, \beta)$ or $IG(v \alpha, \beta)$	Inverted gamma distribution
$\theta \sim p(\theta)$	$\theta$ is a random sample from $p(\theta)$
$\mathbf{a} = [a_1 \ a_2 \ \dots \ a_P]^T$	Vector of autoregressive parameters
$\mathbf{e} = [e_P \ e_{P+1} \ \dots \ e_{N-1}]^T$	Vector of autoregressive excitation samples
$\sigma_e^2$	Variance of excitation sequence
$\sigma_{v_t}^2$	Variance of $t$ th observation noise sample
$\mathbf{y} = [y_0 \ y_1 \ \dots \ y_{N-1}]^T$	Observed (noisy) data vector
$\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]^T$	Underlying signal vector
$\mathbf{i} = [i_0 \ i_1 \ \dots \ i_{N-1}]^T$	Vector of binary noise indicator values
$\mathbf{I}$	The identity matrix
$\mathbf{0}_n$	All zero column vector, length $n$
$\mathbf{0}_{n \times m}$	All zero $(n \times m)$ -dimensional matrix
$\mathbf{1}_n$	All unity column vector, length $n$
$\text{trace}()$	Trace of a matrix
$T$	Transpose of a matrix
$A = \{a_1, \dots, a_M\}$	the set $A$ , containing $M$ elements
$A \cup B$	Union
$A \cap B$	Intersection
$A^c$	Complement
$\emptyset$	Empty set
$a \in A$	$a$ is a member of set $A$
$A \subset B$	$A$ is a subset of $B$
$[a, b]$	real numbers $x$ such that $a \leq x \leq b$
$(a, b)$	real numbers $x$ such that $a < x < b$
$\Re$	the real numbers: $\Re = (-\infty, +\infty)$
$\mathbb{Z}$	the integers: $\{-\infty, \dots, -1, 0, 1, \dots, \infty\}$
$\{\omega : E\}$	'All $\omega$ 's such that expression $E$ is True'

---



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The History of recording – a brief overview . . . . .	2
1.2	Sound restoration – analogue and digital . . . . .	5
1.3	Classes of degradation and overview of the book . . . . .	6
1.4	A reader’s guide . . . . .	10
<b>I</b>	<b>Fundamentals</b>	<b>13</b>
<b>2</b>	<b>Digital Signal Processing</b>	<b>15</b>
2.1	The Nyquist sampling theorem . . . . .	16
2.2	The discrete time Fourier transform (DTFT) . . . . .	19
2.3	Discrete time convolution . . . . .	20
2.4	The $z$ -transform . . . . .	22
2.4.1	Definition . . . . .	23
2.4.2	Transfer function and frequency response . . . . .	23
2.4.3	Poles, zeros and stability . . . . .	23
2.5	Digital filters . . . . .	24
2.5.1	Infinite-impulse-response (IIR) filters . . . . .	24
2.5.2	Finite-impulse-response (FIR) filters . . . . .	25
2.6	The discrete Fourier transform (DFT) . . . . .	25
2.7	Data windows . . . . .	27
2.7.1	Continuous signals . . . . .	27
2.7.2	Discrete-time signals . . . . .	31

2.8	The fast Fourier transform (FFT) . . . . .	34
2.9	Conclusion . . . . .	38
<b>3</b>	<b>Probability Theory and Random Processes</b>	<b>39</b>
3.1	Random events and probability . . . . .	39
3.1.1	Frequency-based interpretation of probability . . . .	40
3.2	Probability spaces . . . . .	40
3.3	Fundamental results of event-based probability . . . . .	42
3.3.1	Conditional probability . . . . .	42
3.3.2	Bayes rule . . . . .	43
3.3.3	Total probability . . . . .	43
3.3.4	Independence . . . . .	44
3.4	Random variables . . . . .	44
3.5	Definition: random variable . . . . .	45
3.6	The probability distribution of a random variable . . . . .	45
3.6.1	Probability mass function (PMF) (discrete RVs) . .	46
3.6.2	Cumulative distribution function (CDF) . . . . .	46
3.6.3	Probability density function (PDF) . . . . .	46
3.7	Conditional distributions and Bayes rule . . . . .	47
3.7.1	Conditional PMF - discrete RVs . . . . .	48
3.7.2	Conditional CDF . . . . .	49
3.7.3	Conditional PDF . . . . .	49
3.8	Expectation . . . . .	52
3.8.1	Characteristic functions . . . . .	53
3.9	Functions of random variables . . . . .	54
3.9.1	Differentiable mappings . . . . .	55
3.10	Random vectors . . . . .	56
3.11	Conditional densities and Bayes rule . . . . .	58
3.12	Functions of random vectors . . . . .	58
3.13	The multivariate Gaussian . . . . .	59
3.14	Random signals . . . . .	61
3.15	Definition: random process . . . . .	64
3.15.1	Mean values and correlation functions . . . . .	64
3.15.2	Stationarity . . . . .	65
3.15.3	Power spectra . . . . .	66
3.15.4	Linear systems and random processes . . . . .	67
3.16	Conclusion . . . . .	67
<b>4</b>	<b>Parameter Estimation, Model Selection and Classification</b>	<b>69</b>
4.1	Parameter estimation . . . . .	70
4.1.1	The general linear model . . . . .	70
4.1.2	Maximum likelihood (ML) estimation . . . . .	71
4.1.3	Bayesian estimation . . . . .	73
4.1.3.1	Posterior inference and Bayesian cost functions . . . . .	75

4.1.3.2	Marginalisation for elimination of unwanted parameters . . . . .	77
4.1.3.3	Choice of priors . . . . .	78
4.1.4	Bayesian Decision theory . . . . .	79
4.1.4.1	Calculation of the evidence, $p(\mathbf{x}   s_i)$ . . . .	80
4.1.4.2	Determination of the MAP state estimate .	82
4.1.5	Sequential Bayesian classification . . . . .	82
4.2	Signal modelling . . . . .	85
4.3	Autoregressive (AR) modelling . . . . .	86
4.3.1	Statistical modelling and estimation of AR models .	87
4.4	State-space models, sequential estimation and the Kalman filter . . . . .	90
4.4.1	The prediction error decomposition . . . . .	91
4.4.2	Relationships with other sequential schemes . . . .	92
4.5	Expectation-maximisation (EM) for MAP estimation . . . .	92
4.6	Markov chain Monte Carlo (MCMC) . . . . .	93
4.7	Conclusions . . . . .	95

## II Basic Restoration Procedures 97

5	Removal of Clicks . . . . .	99
5.1	Modelling of clicks . . . . .	100
5.2	Interpolation of missing samples . . . . .	101
5.2.1	Interpolation for Gaussian signals with known covariance structure . . . . .	103
5.2.1.1	Incorporating a noise model . . . . .	105
5.2.2	Autoregressive (AR) model-based interpolation . . .	106
5.2.2.1	The least squares AR (LSAR) interpolator	106
5.2.2.2	The MAP AR interpolator . . . . .	107
5.2.2.3	Examples of the LSAR interpolator . . . .	108
5.2.2.4	The case of unknown AR model parameters	108
5.2.3	Adaptations to the AR model-based interpolator . .	110
5.2.3.1	Pitch-based extension to the AR interpolator	111
5.2.3.2	Interpolation with an AR + basis function representation . . . . .	111
5.2.3.3	Random sampling methods . . . . .	116
5.2.3.4	Incorporating a noise model . . . . .	119
5.2.3.5	Sequential methods . . . . .	119
5.2.4	ARMA model-based interpolation . . . . .	122
5.2.4.1	Results from the ARMA interpolator . . .	125
5.2.5	Other methods . . . . .	126
5.3	Detection of clicks . . . . .	127
5.3.1	Autoregressive (AR) model-based click detection . .	128
5.3.1.1	Analysis and limitations . . . . .	131

5.3.1.2	Adaptations to the basic AR detector . . .	131
5.3.1.3	Matched filter detector . . . . .	132
5.3.1.4	Other models . . . . .	133
5.4	Statistical methods for the treatment of clicks . . . . .	133
5.5	Discussion . . . . .	134
<b>6</b>	<b>Hiss Reduction</b>	<b>135</b>
6.1	Spectral domain methods . . . . .	136
6.1.1	Noise reduction functions . . . . .	138
6.1.1.1	The Wiener solution . . . . .	139
6.1.1.2	Spectral subtraction and power subtraction	140
6.1.2	Artefacts and ‘musical noise’ . . . . .	141
6.1.3	Improving spectral domain methods . . . . .	145
6.1.3.1	Eliminating musical noise . . . . .	145
6.1.3.2	Advanced noise reduction functions . . . .	148
6.1.3.3	Psychoacoustical methods . . . . .	148
6.1.4	Other transform domain methods . . . . .	148
6.2	Model-based methods . . . . .	149
6.2.1	Discussion . . . . .	149
<b>III</b>	<b>Advanced Topics</b>	<b>151</b>
<b>7</b>	<b>Removal of Low Frequency Noise Pulses</b>	<b>153</b>
7.1	Existing methods . . . . .	154
7.2	Separation of AR processes . . . . .	156
7.3	Restoration of transient noise pulses . . . . .	157
7.3.1	Modified separation algorithm . . . . .	158
7.3.2	Practical considerations . . . . .	160
7.3.2.1	Detection vector $\mathbf{i}$ . . . . .	161
7.3.2.2	AR process for true signal $\mathbf{x}_1$ . . . . .	161
7.3.2.3	AR process for noise transient $\mathbf{x}_2$ . . . . .	162
7.3.3	Experimental evaluation . . . . .	162
7.4	Kalman filter implementation . . . . .	163
7.5	Conclusion . . . . .	163
<b>8</b>	<b>Restoration of Pitch Variation Defects</b>	<b>171</b>
8.1	Overview . . . . .	172
8.2	Frequency tracking . . . . .	175
8.3	Generation of pitch variation curve . . . . .	176
8.3.1	Bayesian estimator . . . . .	178
8.3.2	Prior models . . . . .	180
8.3.2.1	Autoregressive (AR) model . . . . .	180
8.3.2.2	Smoothness Model . . . . .	181

8.3.2.3	Deterministic prior models for the pitch variation . . . . .	182
8.3.3	Discontinuities in frequency tracks . . . . .	182
8.3.4	Experimental results in pitch curve generation . . .	183
8.3.4.1	Trials using extract ‘Viola’ (synthetic pitch variation) . . . . .	184
8.3.4.2	Trials using extract ‘Midsum’ (non-synthetic pitch variation) . . . . .	185
8.4	Restoration of the signal . . . . .	185
8.5	Conclusion . . . . .	190
<b>9</b>	<b>A Bayesian Approach to Click Removal</b>	<b>191</b>
9.1	Modelling framework for click-type degradations . . . . .	192
9.2	Bayesian detection . . . . .	193
9.2.1	Posterior probability for $\mathbf{i}$ . . . . .	194
9.2.2	Detection for autoregressive (AR) processes . . . . .	195
9.2.2.1	Density function for signal, $p_{\mathbf{x}}(\cdot)$ . . . . .	196
9.2.2.2	Density function for noise amplitudes, $p_{\mathbf{n}(\mathbf{i}) \mathbf{i}}(\cdot)$ . . . . .	197
9.2.2.3	Likelihood for Gaussian AR data . . . . .	197
9.2.2.4	Reformulation as a probability ratio test . . . . .	199
9.2.3	Selection of optimal detection state estimate $\mathbf{i}$ . . . . .	200
9.2.4	Computational complexity for the block-based algorithm . . . . .	201
9.3	Extensions to the Bayes detector . . . . .	201
9.3.1	Marginalised distributions . . . . .	201
9.3.2	Noisy data . . . . .	202
9.3.3	Multi-channel detection . . . . .	203
9.3.4	Relationship to existing AR-based detectors . . . . .	203
9.3.5	Sequential Bayes detection . . . . .	203
9.4	Discussion . . . . .	204
<b>10</b>	<b>Bayesian Sequential Click Removal</b>	<b>205</b>
10.1	Overview of the method . . . . .	206
10.2	Recursive update for posterior state probability . . . . .	207
10.2.1	Full update scheme. . . . .	209
10.2.2	Computational complexity . . . . .	210
10.2.3	Kalman filter implementation of the likelihood update . . . . .	210
10.2.4	Choice of noise generator prior $p(\mathbf{i})$ . . . . .	211
10.3	Algorithms for selection of the detection vector . . . . .	211
10.3.1	Alternative risk functions . . . . .	213
10.4	Summary . . . . .	213
<b>11</b>	<b>Implementation and Experimental Results for Bayesian Detection</b>	<b>215</b>
11.1	Block-based detection . . . . .	216

11.1.1	Search procedures for the MAP detection estimate . . . . .	216
11.1.2	Experimental evaluation . . . . .	219
11.2	Sequential detection . . . . .	222
11.2.1	Synthetic AR data . . . . .	225
11.2.2	Real data . . . . .	226
11.3	Conclusion . . . . .	227
<b>12</b>	<b>Fully Bayesian Restoration using EM and MCMC</b>	<b>233</b>
12.1	A review of some relevant work from other fields in Monte Carlo methods . . . . .	234
12.2	Model specification . . . . .	235
12.2.1	Noise specification . . . . .	235
12.2.1.1	Continuous noise source . . . . .	235
12.2.2	Signal specification . . . . .	236
12.3	Priors . . . . .	237
12.3.1	Prior distribution for noise variances . . . . .	237
12.3.2	Prior for detection indicator variables . . . . .	239
12.3.3	Prior for signal model parameters . . . . .	239
12.4	EM algorithm . . . . .	239
12.5	Gibbs sampler . . . . .	241
12.5.1	Interpolation . . . . .	242
12.5.2	Detection . . . . .	244
12.5.3	Detection in sub-blocks . . . . .	246
12.6	Results for EM and Gibbs sampler interpolation . . . . .	248
12.7	Implementation of Gibbs sampler detection/interpolation . . . . .	249
12.7.1	Sampling scheme . . . . .	249
12.7.2	Computational requirements . . . . .	251
12.7.2.1	Comparison of computation with existing techniques . . . . .	253
12.8	Results for Gibbs sampler detection/interpolation . . . . .	254
12.8.1	Evaluation with synthetic data . . . . .	254
12.8.2	Evaluation with real data . . . . .	256
12.8.3	Robust initialisation . . . . .	257
12.8.4	Processing for audio evaluation . . . . .	258
12.8.5	Discussion of MCMC applied to audio . . . . .	259
<b>13</b>	<b>Summary and Future Research Directions</b>	<b>271</b>
13.1	Future directions and new areas . . . . .	272
<b>A</b>	<b>Probability Densities and Integrals</b>	<b>275</b>
A.1	Univariate Gaussian . . . . .	275
A.2	Multivariate Gaussian . . . . .	275
A.3	Gamma density . . . . .	277
A.4	Inverted Gamma distribution . . . . .	277



<b>B</b>	<b>Matrix Inverse Updating Results and Associated Properties</b>	<b>279</b>
<b>C</b>	<b>Exact Likelihood for AR Process</b>	<b>281</b>
<b>D</b>	<b>Derivation of Likelihood for <math>\mathbf{i}</math></b>	<b>283</b>
D.1	Gaussian noise bursts . . . . .	284
<b>E</b>	<b>Marginalised Bayesian Detector</b>	<b>285</b>
<b>F</b>	<b>Derivation of Sequential Update Formulae</b>	<b>287</b>
F.1	Update for $i_{n+1} = 0$ . . . . .	287
F.2	Update for $i_{n+1} = 1$ . . . . .	289
<b>G</b>	<b>Derivations for EM-based Interpolation</b>	<b>293</b>
G.1	Signal expectation, $E_x$ . . . . .	294
G.2	Noise expectation, $E_n$ . . . . .	296
G.3	Maximisation step . . . . .	297
<b>H</b>	<b>Derivations for Gibbs Sampler</b>	<b>299</b>
H.1	Gaussian/inverted-gamma scale mixtures . . . . .	299
H.2	Posterior distributions . . . . .	300
H.2.1	Joint posterior . . . . .	300
H.2.2	Conditional posteriors . . . . .	302
H.3	Sampling the prior hyperparameters . . . . .	303
	<b>References</b>	<b>305</b>
	<b>Index</b>	<b>321</b>



# 1

## Introduction

The introduction of high quality digital audio media such as Compact Disc (CD) and Digital Audio Tape (DAT) has dramatically raised general awareness and expectations about sound quality in all types of recordings. This, combined with an upsurge of interest in historical and nostalgic material, has led to a growing requirement for the restoration of degraded sources ranging from the earliest recordings made on wax cylinders in the nineteenth century, through disc recordings (78rpm, LP, etc.) and finally magnetic tape recording technology, which has been available since the 1950s. Noise reduction may occasionally be required even in a contemporary digital recording if background noise is judged to be intrusive.

Degradation of an audio source will be considered as any undesirable modification to the audio signal which occurs as a result of (or subsequent to) the recording process. For example, in a recording made direct-to-disc from a microphone, degradations could include noise in the microphone and amplifier as well as noise in the disc cutting process. Further noise may be introduced by imperfections in the pressing material, transcription to other media or wear and tear of the medium itself. Examples of such noise can be seen in the electron micrographs shown in figures 1.1-1.3. In figure 1.1 we can clearly see the specks of dust on the groove walls and also the granularity in the pressing material, which can be seen sticking out of the walls. In figure 1.2 the groove wall signal modulation can be more clearly seen. In figure 1.3, a broken record is seen almost end-on. Note the fragments of

FIGURE 1.1. Electron micrograph showing dust and granular particles in the grooves of a 78rpm gramophone disc.

broken disk which fill the grooves<sup>1</sup>. We do not strictly consider any noise present in the recording environment such as audience noise at a musical performance to be degradation, since this is part of the ‘performance’. Removal of such performance interference is a related topic which is considered in other applications, such as speaker separation for hearing aid design. An ideal restoration would then reconstruct the original sound source exactly as received by the transducing equipment (microphone, acoustic horn, etc.). Of course, this ideal can never be achieved perfectly in practice, and methods can only be devised which come close according to some suitable error criterion. This should ideally be based on the perceptual characteristics of the human listener.

## 1.1 The History of recording – a brief overview

The ideas behind sound recording<sup>2</sup> began as early as the mid-nineteenth century with the invention by Frenchman Léon Scott of the Phonautograph in 1857, a device which could display voice waveforms on a piece of paper

---

<sup>1</sup>These photographs are reproduced with acknowledgement to Mr. B.C. Breton, Scientific Imaging Group, CUED

<sup>2</sup>For a more detailed coverage of the origins of sound recording see, for example, Peter Martland’s excellent book ‘Since Records Began: EMI, the First Hundred Years’[124].

FIGURE 1.2. Electron micrograph showing signal modulation of the grooves of a 78rpm gramophone disc.

FIGURE 1.3. Electron micrograph showing a breakage directly across a 78rpm gramophone disc.

via a recording horn which focussed sound onto a vibrating diaphragm. It was not until 1877, however, that Thomas Edison invented the Phonograph, a machine capable not only of storing but of reproducing sound, using a steel point stylus which cut into a tin foil rotating drum recording medium. This piece of equipment was a laboratory tool rather than a commercial product, but by 1885 Alexander Graham Bell, and two partners, C.A. Bell and C.S. Tainter, had developed the technology sufficiently to make it a commercial proposition, and along the way had experimented with technologies which would later become the gramophone disc, magnetic tape and optical soundtracks. The new technology worked by cutting into a beeswax cylinder, a method originally designed to bypass the 1878 patent of Edison. In 1888, commercial exploitation of this new wax cylinder technology began, the same year that Emile Berliner demonstrated the first flat disc record and gramophone at the Franklin Institute, using an acid etching process to cut grooves into the surface of a polished zinc plate.

Cylinder technology was cumbersome in the extreme, and no cheap method for duplicating cylinders became available until 1902. This meant that artists had to perform the same pieces many times into multiple recording horns in order to obtain a sufficient quantity for sale. Meanwhile Berliner developed the gramophone disc technology, discovering that shellac, a material derived from the Lac beetle, was a suitable medium for manufacturing records from his metal negative master discs. Shellac was used for 78rpm recordings until vinyl was invented in the middle of this century. By 1895 a catalogue of a hundred 7-inch records had been produced by the Berliner Gramophone Company, but the equipment was hand-cranked and primitive. Further developments led to a motor driven gramophone, more sensitive soundbox and a new wax disc recording process, which enabled the gramophone to become a huge commercial success.

Recording using a mechanical horn and diaphragm was a difficult and unreliable procedure, requiring performers to crowd around the recording horn in order to be heard, and some instruments could not be adequately recorded at all, owing to the poor frequency response of the system (164Hz–2000Hz). The next big development was the introduction in 1925, after various experiments from 1920 onwards, of the Western Electric electrical recording process into the Columbia studios, involving an electrical microphone and amplifier which actuates a cutting tool. The electric process led to great improvements in sound quality, including a bandwidth of 20Hz–5000Hz and reduced surface noise. The technology was much improved by Alan Blumlein, who also demonstrated the first stereo recording process in 1935. In the same year AEG in Germany demonstrated the precursor of the magnetic tape recording system, a method which eliminates the clicks, crackles and pops of disc recordings. Tape recording was developed to its modern form by 1947, allowing for the first time a practical way to edit recordings, including of course to ‘cut and splice’ the tape for restoration purposes. In the late forties the vinyl LP and 45rpm single were launched.

In 1954 stereophonic tapes were first manufactured and in 1958 the first stereo discs were cut. Around this time analogue electronic technology for sound modification by filtering, limiting, compressing and equalisation was being introduced, which allowed the filtering of recordings for reduction of surface noise and enhancement of selective frequencies.

The revolution which has allowed the digital processing techniques of this book to succeed is the introduction of digital sound, in particular in 1982 the compact disc (CD) format, a digital format which allows a stereo signal bandwidth up to 20kHz with 16-bit resolution. Now of course we see higher bandwidth (48kHz), better resolution (24-bit) formats being used in the recording studio, but the CD has proved itself as the first practical domestic digital format.

## 1.2 Sound restoration – analogue and digital

Analogue restoration techniques have been available for at least as long as magnetic tape, in the form of manual cut-and-splice editing for clicks and frequency domain equalisation for background noise (early mechanical disc playback equipment will also have this effect by virtue of its poor response at high frequencies). More sophisticated electronic click reducers were based upon high pass filtering for detection of clicks, and low pass filtering to mask their effect (see e.g. [34, 102])<sup>3</sup>. None of these methods was sophisticated enough to perform a significant degree of noise reduction without interfering with the underlying signal quality. For analogue tape recordings the pre-emphasis techniques of Dolby [47] have been very successful in reducing the levels of background noise in analogue tape, but of course the pre-emphasis has to be encoded into the signal at the recording stages.

Digital methods allow for a much greater degree of flexibility in processing, and hence greater potential for noise removal, although indiscriminate application of inappropriate digital methods can be more disastrous than analogue processing! Some of the earliest digital signal processing work for audio restoration involved deconvolution for enhancement of a solo voice (Caruso) from an acoustically recorded source (see Miller [132] and Stockham *et al.* [171]). Since then, research groups from many places, including Cambridge, Le Mans, Paris and the US, have worked in the area, developing sophisticated techniques for treatment of degraded audio. For a good overall text on the field of digital audio, including restoration [84], see [25]. Another text which covers many enhancement techniques related to those of this book is by Vaseghi [188]. On the commercial side, academic research has led to spin-off companies which manufacture computer

---

<sup>3</sup>The well known ‘Packburn’ unit achieved masking within a stereo setup by switching between channels

restoration equipment for use in recording studios, re-mastering houses and broadcast companies. In Cambridge, CEDAR (Computer Enhanced Digital Audio Restoration) Ltd., founded in 1988 by Dr. Peter Rayner from the Communications Group of the University's Engineering Department and the British Library's National Sound Archive, is probably the best known of these companies, providing equipment for automatic real-time noise reduction, click and crackle removal, while the California-based Sonic Solutions markets a well-known system called NoNoise. Now, with the rapid increases in cheaply available computer performance, many computer editing packages include noise and click reduction as a standard add-on.

### 1.3 Classes of degradation and overview of the book

There are several distinct types of degradation common in audio sources. These can be broadly classified into two groups: *localised* degradations and *global* degradations. Localised degradations are discontinuities in the waveform which affect only certain samples, including clicks, crackles, scratches, breakages and clipping. Global degradations affect all samples of the waveform and include background noise, wow and flutter and certain types of non-linear distortion. Mechanisms by which all of these defects can occur are discussed later. We distinguish the following classes of localised degradation:

- Clicks - these are short bursts of interference random in time and amplitude. Clicks are perceived as a variety of defects ranging from isolated 'tick' noises to the characteristic 'crackle' associated with 78rpm disc recordings.
- Low frequency noise transients - usually a larger scale defect than clicks and caused by very large scratches or breakages in the playback medium. These large discontinuities excite a low frequency resonance in the pickup apparatus which is perceived as a low frequency 'thump' noise. This type of degradation is common in gramophone disc recordings and optical film sound tracks.

Global degradations affect all samples of the waveform and the following categories may be identified:

- Broad band noise - this form of degradation is common to all recording methods and is perceived as 'hiss'.
- Wow and flutter - these are pitch variation defects which may be caused by eccentricities in the playback system or motor speed fluctuations. The effect is a very disturbing modulation of all frequency components.



- Distortion - This very general class covers a wide range of non-linear defects from amplitude related overload (e.g. clipping) to groove wall deformation and tracing distortion.

We describe methods in this text to address the majority of the above defects (in fact all except non-linear distortion, which is a topic of current research interest in DSP for audio). In the case of localised degradations a major task is the detection of discontinuities in the waveform. In section III we adopt a classification approach to this task which is based on Bayes Decision Theory. Using a probabilistic model-based formulation for the audio data and degradation we use Bayes' theorem to derive optimal detection estimates for the discontinuities. In the case of global degradations an approach based on probabilistic noise and data models is applied to give estimates for the true (undistorted) data conditional on the observed (noisy) data.

Note that all audio results and experimentation presented are performed using audio signals sampled at the professional rates of 44.1kHz or 48kHz and quantised to 16 bits.

The following gives a brief outline of the ensuing chapters.

## Part I - Fundamentals

In the first part of the book we provide an overview of the basic theory on which the rest of the book relies. These chapters are not intended to be a complete tutorial for a reader completely unfamiliar with the area, but rather a summary of the important results in a form which is easily accessible. The reader is assumed to be quite familiar with linear systems theory and continuous time spectral analysis. Much of the material in this first section is based upon courses taught by the authors to undergraduates at Cambridge University.

### *Chapter 2 - Digital Signal Processing*

In this chapter we overview the basics of digital signal processing (DSP), the theory of discrete time processing of sampled signals. We include an introduction to sampling theory, convolution, spectrum analysis, the  $z$ -transform and digital filters.

### *Chapter 3 - Probability Theory and Random Processes*

Owing to the random nature of most audio signals it is necessary to have a thorough grounding in random signal theory in order to design effective restoration methods. Indeed, most of the restoration methods presented in the text are based explicitly on probabilistic models for signals and

noise. In this chapter we build up the theory of random processes, including correlation functions, power spectra and linear systems analysis from the fundamentals of probability theory for random variables and random vectors.

### *Chapter 4 - Parameter Estimation, Classification and Model Selection*

This chapter introduces the fundamental concepts and techniques of parameter estimation and model selection, topics which are applied throughout the text, with an emphasis upon the Bayesian Decision Theory perspective. A mathematical framework based on a linear parameter Gaussian data model is used throughout to illustrate the methods. We then consider some basic signal models which will be of use later in the text and describe some powerful numerical statistical estimation methods: expectation-maximisation (EM) and Markov chain Monte Carlo (MCMC).

## Part II - Basic Restoration Procedures

Part II describes the basic methods for removing clicks and background noise from disc and tape recordings. Much of the material is a review of methods which might be used in commercial re-mastering environments, however we also include some new material such as interpolation using the autoregressive moving-average (ARMA) models. The reader who is new to the topic of digital audio restoration will find this part of the book an invaluable introduction to the wide range of methods which can be applied.

### *Chapter 5 - Removal of Clicks*

Clicks are the most common form of artefact encountered in old recordings and the first stage of processing in many cases will be a de-clicking process. We describe firstly a range of techniques for interpolation of missing samples in audio; this is the task required for replacing a click in the audio waveform. We then discuss methods for detection of clicks, based upon modelling the distinguishing features of audio signals and abrupt discontinuities in the waveform (clicks). The methods are illustrated throughout with graphical examples which contrast the performance of the various schemes.

### *Chapter 6 - Hiss Reduction*

All recordings, whatever their source, are inherently contaminated with some degree of background noise, often perceived as ‘hiss’. This chapter

describes the technology for hiss reduction, mostly based upon a frequency domain attenuation principle. We then go on to describe how hiss reduction can be carried out in a model-based setting.

## Part III - Advanced Topics

In this section we describe some more recent and active research topics in audio restoration. The research spans the period 1990-1998 and can be considered to form a ‘second generation’ of sophisticated techniques which can handle new problems such as ‘wow’ or potentially achieve improvements in the basic areas such as click removal. These methods are generally not implemented in the commercial systems of today, but are likely to form a part of future systems as computers become faster and cheaper. Many of the ideas for these research topics have come from the authors’ experience in the commercial sound restoration arena.

### *Chapter 7 - Removal of Low Frequency Noise Pulses*

Low frequency noise pulses occur in gramophone and optical film media when the playback system is driven by step-like inputs such as groove break-ages or large scratches. We firstly review the existing template-based methods for restoration of such defects, then present a new signal separation-based approach in which both the audio signal and the noise pulse are modelled as autoregressive (AR) processes. A separation algorithm is developed which removes the noise pulse from the observed signal. The algorithm has more general application than existing methods which rely on a ‘template’ for the noise pulse. Performance is found to be better than the existing methods and the new process is more readily automated.

### *Chapter 8 - Restoration of Pitch Variation Defects*

This chapter presents a novel technique for restoration of musical material degraded by wow and other related pitch variation defects. An initial frequency tracking stage extracts frequency tracks for all significant tonal components of the music. This is followed by an estimation procedure which identifies pitch variations which are common to all components, under the assumption of smooth pitch variation with time. Restoration is then performed by non-uniform resampling of the distorted signal. Results show that wow can be virtually eliminated from musical material which has a significant tonal component.

*Chapters 9-11 - A Bayesian Approach to Click Removal*

In these chapters a new approach to click detection and replacement is developed. This approach is based upon Bayes decision theory as discussed in chapter 4. A Gaussian autoregressive (AR) model is assumed for the data and a Gaussian model is also used for click amplitudes. The detector is shown to be equivalent under certain limiting constraints to the existing AR model-based detectors currently used in audio restoration. In chapter 10 a novel sequential implementation of the Bayesian techniques is developed and in chapter 11 results are presented demonstrating that the new methods can out-perform the methods described in chapter 5.

*Chapter 12 - Fully Bayesian Restoration using EM and MCMC*

The Bayesian methods of chapters 9-11 led to improvements in detection and restoration of clicks. However, a disadvantage is that system parameters must be known *a priori* or estimated in some *ad hoc* fashion from the data. In this chapter we present more sophisticated statistical methodology for solution of these limitations and for more realistic modelling of signal and noise sources. We firstly present an expectation-maximisation (EM) method for interpolation of autoregressive signals in non-Gaussian impulsive noise. We then present a Markov chain Monte Carlo (MCMC) technique which is capable of performing interpolation *jointly* with detection of clicks. This is a significant advance and the drawback is a dramatic increase in computational complexity for the algorithm. However, we believe that with the rapid advances in computational power which are constantly occurring, methods such as these will come to dominate complex statistical signal processing problem-solving in the future. The chapter provides an in-depth case study of EM and MCMC applied to click removal, but it is noted that the methods can be applied with benefit to many of the other problem areas described in the book.

## 1.4 A reader's guide

This book is aimed at a wide range of readers, from the technically minded audio enthusiast through to research scientists in industrial and academic environments. For those who have little knowledge of statistical signal processing the introductory chapters in Section I will be essential reading, and it may be necessary to refer to some of the cited texts in addition as the coverage is of necessity rather sparse. Part II will then provide the core reading material on restoration techniques, with Part III providing some interesting developments into areas such as wow and low frequency pulse removal. For the reader with a research background in statistical signal processing, Part I will serve only as a reference for notation and terminol-

ogy, although chapter 4 may provide some useful insights into the Bayesian methodology adopted for much of the book. Part II will then provide general background reading in basic restoration methods, leading to Part III which contains state-of-the-art research in the audio restoration area.

Chapters 5 and 9-12 may be read in conjunction for those interested in click and crackle treatment techniques, while chapters 6, 7 and 8 form stand-alone texts on the areas of hiss reduction, low frequency pulse removal and wow removal, respectively.



# Part I

## Fundamentals





## 2

# Digital Signal Processing

Digital signal processing (DSP) is a technique for implementing operations such as signal filtering and spectrum analysis in digital form, as shown in the block diagram of figure 2.1.

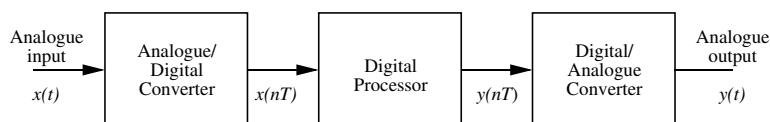


FIGURE 2.1. Digital signal processing system

There are many advantages in carrying out digital rather than analogue processing; amongst these are flexibility and repeatability. The flexibility stems from the fact that system parameters are simply numbers stored in the processor. Thus for example, it is a trivial matter to change the cut-off frequency of a digital filter whereas a lumped element analogue filter would require a different set of passive components. Indeed the ease with which system parameters can be changed has led to many adaptive techniques whereby the system parameters are modified in real time according to some algorithm. Examples of this are adaptive equalisation of transmission systems and adaptive antenna arrays which automatically steer the nulls in the polar diagram onto interfering signals. Digital signal processing enables very complex linear and non-linear processes to be implemented which would not be feasible with analogue processing. For example it is

difficult to envisage an analogue system which could be used to perform spatial filtering of an image to improve the signal to noise ratio.

DSP has been an active research area since the late 1960s but applications tended to be only in large and expensive systems or in non-real time where a general purpose computer could be used. However, the advent of DSP chips has enabled real-time processing to be performed at very low cost and has enabled audio companies such as CEDAR Audio and Sonic Solutions to produce real-time restoration systems for remastering studios and record companies. The next few years will see the further integration of DSP into domestic products such as television, radio, mobile telephones and of course hi-fi equipment.

In this chapter we review the basic theory of digital signal processing (DSP) as required later in the book. We consider firstly Nyquist sampling theory, which states that a continuous time signal such as an audio signal which is band-limited in frequency can be represented perfectly without any information loss as a set of discrete digital sample values. The whole of digital audio, including compact disc (CD) and digital audio tape (DAT) relies heavily on the validity of this theory and so a strong understanding is essential. The chapter then proceeds to describe signal manipulation in the digital domain, covering such important topics as Fourier analysis and the discrete Fourier transform (DFT), the  $z$ -transform, time domain data windows and the fast Fourier transform (FFT). A basic level of knowledge in continuous time linear systems, Laplace and Fourier analysis is assumed. Much of the material is of necessity given a superficial coverage and for more detailed descriptions the reader is referred to the texts by Roberts and Mullis [164], Proakis and Manolakis [157] and Oppenheim and Schaffer [145].

## 2.1 The Nyquist sampling theorem

It is necessary to determine under what conditions a continuous signal  $g(t)$  may be unambiguously represented by a series of samples taken from the signal at uniform intervals of  $T$ . It is convenient to represent the sampling process as that of multiplying the continuous signal  $g(t)$  by a sampling signal  $s(t)$  which is an infinite train of impulse functions  $\delta(nT)$ . The sampled signal  $g_s(t)$  is:

$$g_s(t) = g(t) s(t)$$

where:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Now the impulse train  $s(t)$  is a periodic function and may be represented by a Fourier series:

$$s(t) = \sum_{p=-\infty}^{\infty} c_p e^{jp\omega_0 t}$$

where

$$c_p = \frac{1}{T} \int_{-\frac{T}{2}}^{-\frac{T}{2}} s(t) e^{-jp\omega_0 t} dt = \frac{1}{T}$$

and

$$\omega_0 = \frac{2\pi}{T}$$

$$\therefore g_s(t) = g(t) \frac{1}{T} \sum_{p=-\infty}^{\infty} e^{jp\omega_0 t}$$

The spectrum  $G_s(\omega)$  of the sampled signal may be determined by taking the Fourier transform of  $g_s(t)$ . This is most readily achieved by making use of the frequency shift theorem which states that:

$$\text{If } g(t) \Rightarrow G(\omega)$$

$$\text{then } g(t) e^{j\omega_0 t} \Rightarrow G(\omega - \omega_0)$$

Application of this theorem gives:

$$\boxed{G_s(\omega) = \frac{1}{T} \sum_{p=-\infty}^{\infty} G(\omega - p\omega_0)} \quad (2.1)$$

*Spectrum of sampled signal*

The above equation shows that the spectrum of the sampled signal is simply the sum of the spectra of the continuous signal repeated periodically at intervals of  $\omega_0 = \frac{2\pi}{T}$  as shown in figure 2.2. Thus the continuous signal  $g(t)$  may be perfectly recovered from the sampled signal  $g_s(t)$  provided that the sampling interval  $T$  is chosen so that:

$$\frac{2\pi}{T} > 2 \omega_B$$

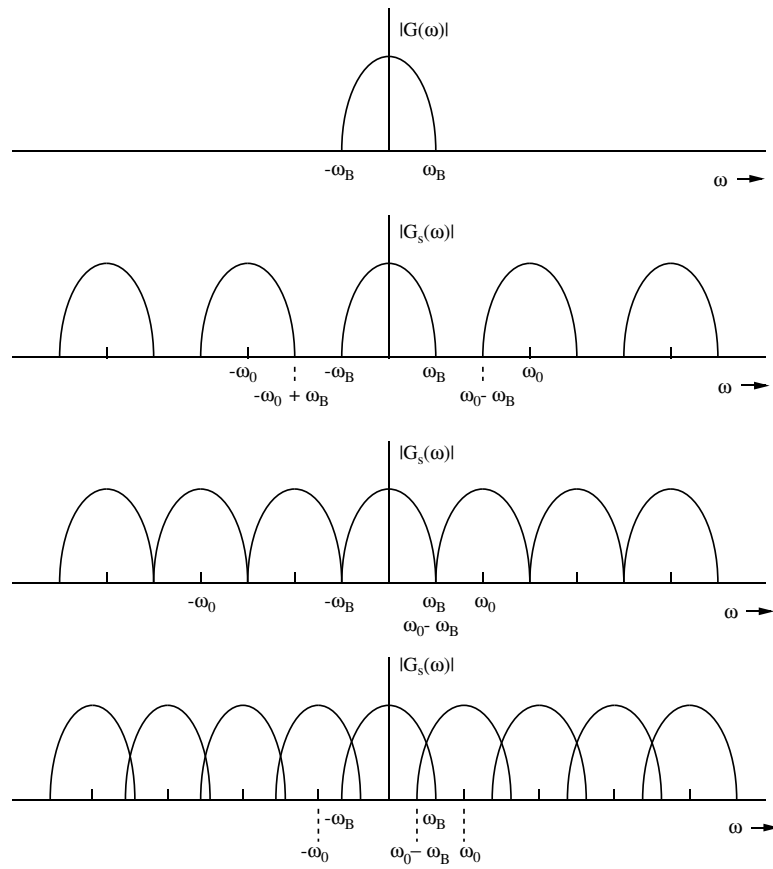


FIGURE 2.2. Sampled signal spectra for various different sampling frequencies  $\omega_0$

where  $\omega_B$  is the bandwidth of the continuous signal  $g(t)$ . If this condition holds then there is no overlap of the individual spectra in the summation of expression (2.1) and the original signal can be perfectly reconstructed using a low-pass filter  $H(\omega)$  with bandwidth  $\omega_B$ :

$$H(\omega) = \begin{cases} 1, & |\omega| < \omega_B \\ 0, & \text{Otherwise} \end{cases}$$

This theory underpins the whole of digital audio and means that a digital audio system can in principle be designed which loses none of the information contained in the continuous domain signal  $g(t)$ . Of course the practicalities are rather different as a band-limited input signal is assumed and the reconstruction filter  $H(\omega)$ , the A/D and D/A converters must be ideal. For more detail of this procedure see for example [99]

## 2.2 The discrete time Fourier transform (DTFT)

We now proceed to calculate the sampled signal spectrum in an alternative way which leads to the discrete time Fourier transform (DTFT). The sampled signal  $g_s(t)$  is given by:

$$g_s(t) = g(t) \sum_{p=-\infty}^{\infty} \delta(t - pT)$$

and the signal spectrum  $G_s(\omega)$  can be obtained by taking the Fourier transform directly:

$$\begin{aligned} G_s(\omega) &= \int_{-\infty}^{\infty} g_s(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} g(t) \sum_{p=-\infty}^{\infty} \delta(t - pT) e^{-j\omega t} dt \\ \therefore G_s(\omega) &= \sum_{p=-\infty}^{\infty} g_p e^{-j\omega pT} \end{aligned} \quad (2.2)$$

where we have defined  $g_p = g(pT)$ . Note that this is a periodic function of frequency and is usually written in the following form:

$$\boxed{G(e^{j\omega T}) = \sum_{p=-\infty}^{\infty} g_p e^{-j\omega pT}} \quad (2.3)$$

*Discrete time Fourier transform (DTFT)*

The signal sample values may be expressed in terms of the sampled signal spectrum by noting that equation (2.3) has the form of a Fourier series and the orthogonality of the complex exponential can be used to invert the transform. Multiply each side of equation (2.3) by  $e^{jn\omega T}$  and integrate:

$$\begin{aligned}\int_0^{2\pi} G(e^{j\omega T}) e^{jn\omega T} d\omega T &= \int_0^{2\pi} \sum_{p=-\infty}^{\infty} g_p e^{-j\omega(p-n)T} d\omega T \\ &= \sum_{p=-\infty}^{\infty} g_p \int_0^{2\pi} e^{-j\omega(p-n)T} d\omega T\end{aligned}$$

but

$$\int_0^{2\pi} e^{-j(p-n)\omega T} d\omega T = \begin{cases} 0, & p \neq n \\ 2\pi, & p = n \end{cases}$$

The signal samples are thus obtained from the DTFT as:

$$\boxed{g_n = \frac{1}{2\pi} \int_0^{2\pi} G(e^{j\omega T}) e^{jn\omega T} d\omega T} \quad (2.4)$$

*Inverse DTFT*

## 2.3 Discrete time convolution

Consider an analogue signal  $x(t)$  and its sampled representation

$$x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) = \sum_{n=-\infty}^{\infty} x(nT) \delta(t - nT).$$

If we apply  $x_s(t)$  as the input to a linear time invariant (LTI) system with impulse response  $h(t)$ , then the output signal can be written as

$$\begin{aligned}y(t) &= \int_{-\infty}^{\infty} h(t - \tau) \sum_{m=-\infty}^{\infty} x(mT) \delta(\tau - mT) d\tau \\ &= \sum_{m=-\infty}^{\infty} h(t - mT) x(mT)\end{aligned}$$

If we now evaluate the output at times  $nT$  we have:

$$y(nT) = \sum_{m=-\infty}^{\infty} h((n-m)T)x(mT) = \sum_{m=-\infty}^{\infty} h(mT)x((n-m)T) \quad (2.5)$$

in which the impulse response needs only to be evaluated at integer multiples of  $T$ . This discrete convolution can be evaluated in the digital domain as shown in figure 2.3, where  $x(nT)$  represents values from an analogue signal  $x(t)$  sampled periodically at intervals of  $T$  and the outputs  $y(nT)$  represent sample values which may be converted into an analogue signal by means of a digital to analogue converter. We will denote the digitised sequences from now on as  $x_n = x(nT)$  and  $y_n = y(nT)$ , and the sampled impulse response as  $h_n = h(nT)$ , leading to the discrete time convolution equations:

$$y_n = \sum_{m=-\infty}^{\infty} h_{n-m}x_m = \sum_{m=-\infty}^{\infty} h_mx_{n-m} \quad (2.6)$$

*Discrete time convolution*

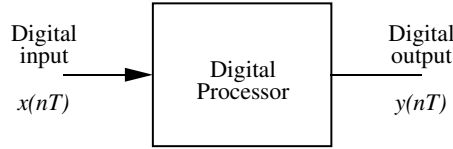


FIGURE 2.3. Digital Signal Processor

The frequency domain relationship between the system input and output may be developed from the convolution relationship as follows. Take the DTFT of each side of equation (2.6):

$$\begin{aligned} Y(e^{j\omega T}) &= \sum_{n=-\infty}^{\infty} y_n e^{-jn\omega T} = \sum_{n=-\infty}^{\infty} \left\{ \sum_{p=-\infty}^{\infty} x_p h_{n-p} \right\} e^{-jn\omega T} \\ \therefore Y(e^{j\omega T}) &= \sum_{p=-\infty}^{\infty} x_p \sum_{n=-\infty}^{\infty} h_{n-p} e^{-jn\omega T} \end{aligned}$$

Let  $n - p = q$ , then

$$\begin{aligned}
Y(e^{j\omega T}) &= \sum_{p=-\infty}^{\infty} x_p \sum_{q=-\infty}^{\infty} h_q e^{-(q+p)\omega T} \\
&= \left\{ \sum_{p=-\infty}^{\infty} x_p e^{-jp\omega T} \right\} \left\{ \sum_{q=-\infty}^{\infty} h_q e^{-jq\omega T} \right\}
\end{aligned}$$

But  $\sum_{p=-\infty}^{\infty} x_p e^{-jp\omega T}$  is the spectrum of the system input,  $X(e^{j\omega T})$ ,

$$\boxed{\therefore Y(e^{j\omega T}) = X(e^{j\omega T}) H(e^{j\omega T})} \quad (2.7)$$

where

$$\boxed{H(e^{j\omega T}) = \sum_{q=-\infty}^{\infty} h_q \cdot e^{-jq\omega T}} \quad (2.8)$$

is the system frequency response.

The system frequency response could also have been derived directly from equation (2.6) as follows. Let

$$\begin{aligned}
x_p &= e^{j\omega p T} \\
\therefore y_n &= \sum_{p=-\infty}^{\infty} h_p \cdot e^{j\omega(n-p)T} \\
\therefore y_n &= e^{j\omega n T} \sum_{p=-\infty}^{\infty} h_p \cdot e^{-jp\omega T} \\
\therefore H(e^{j\omega T}) &\equiv \sum_{p=-\infty}^{\infty} h_p e^{-jp\omega T}
\end{aligned}$$

## 2.4 The $z$ -transform

The Laplace transform is an important tool in continuous system theory as it enables one to deal with differential equations as algebraic equations, since it transforms convolutive functions into multiplicative functions. However, the natural mathematical description of sampled data systems is in terms of difference equations and it would be of considerable help to develop a method whereby difference equations can be treated as algebraic equations. The  $z$ -transform accomplishes this and is also a powerful tool for general interpretation of discrete system behaviour.



### 2.4.1 Definition

The two-sided  $z$ -transform is defined for a sampled signal  $\{g_n\}$  as:

$$G(z) = \mathcal{Z}\{g_n\} = \sum_{p=-\infty}^{\infty} g_p z^{-p} \quad (2.9)$$

*2-sided  $z$ -transform*

The  $z$ -transform exists for all  $z$  such that the summation converges absolutely, i.e.

$$\sum_{p=-\infty}^{\infty} |g_p z^{-p}| < \infty$$

### 2.4.2 Transfer function and frequency response

We can see a similarity here with the DTFT, equation (2.3), in that the two-sided  $z$ -transform is equal to the DTFT when  $z = e^{j\omega T}$ . Equivalent results can be shown to apply for the discrete-time convolution and frequency response in the  $z$ -domain; specifically, if we obtain the  $z$ -transform of the discrete time impulse response,  $H(z) = \sum_{p=-\infty}^{\infty} h_p z^{-p}$ , then the input-output relationships in the time domain and the  $z$ -domain are:

$$y_n = \sum_{m=-\infty}^{\infty} h_m x_{n-m} \text{ (time domain)} \Leftrightarrow Y(z) = H(z) X(z) \text{ (} z\text{-domain)}$$

where  $H(z)$  is known as the transfer function and  $Y(z)$  and  $X(z)$  are the  $z$ -transforms of the output and input sequences, respectively. The frequency response of the system is:

$$H(z) \Big|_{z=e^{j\omega T}}$$

Another useful result is the time-shifting theorem for  $z$ -transforms, which is used for handling difference equations and digital filters:

$$\mathcal{Z}\{g_{n-p}\} = z^{-p} G(z) \quad (2.10)$$

### 2.4.3 Poles, zeros and stability

If we suppose that the transfer function is a rational function of  $z$ , i.e. the ratio of two polynomials in  $z$ , then  $H(z)$  can be factorised as:

$$H(z) = K \frac{\prod_{i=1}^M (z - z_i)}{\prod_{j=1}^N (z - p_j)}$$

in which the  $\{z_i\}$  are known as the *zeros* and  $\{p_i\}$  as the *poles* of the transfer function. Then, a necessary and sufficient condition for the transfer function to be bounded-input bounded-output (BIBO) stable<sup>1</sup> is that all the poles lie within the unit circle, i.e.

$$|p_i| < 1, \quad i = 1, \dots, N$$

There are many more mathematical and practical details of the  $z$ -transform, but we omit these here as they are not used elsewhere in the text. The interested reader is referred to [164] or one of the many other excellent texts on the topic.

## 2.5 Digital filters

### 2.5.1 Infinite-impulse-response (IIR) filters

Consider a discrete time difference equation input-output relationship of the form:

$$y_n = \sum_{i=0}^M b_i x_{n-i} + \sum_{j=1}^N a_j y_{n-j}$$

This can be straightforwardly implemented in hardware or software using a network of delays, adders and multipliers and is known as an *infinite-impulse-response* (IIR) filter. Taking  $z$ -transforms of both sides using result (2.10) and rearranging, we obtain:

$$Y(z) = \frac{B(z)}{A(z)} X(z)$$

where

$$B(z) = \sum_{i=0}^M b_i z^{-i}$$

and

$$A(z) = 1 - \sum_{j=1}^N a_j z^{-j}$$

The transfer function is  $H(z) = \frac{B(z)}{A(z)}$ , which is stable if and only if the zeros of  $A(z)$  lie within the unit circle. Furthermore, the filter is *invertible* (this

---

<sup>1</sup>This means that no finite-amplitude input sequence  $\{x_n\}$  can generate an infinite-amplitude output

means that a finite sequence  $\{x_n\}$  can always be obtained from any finite amplitude sequence  $\{y_n\}$  if and only if the zeros of  $B(z)$  also lie within the unit circle. This type of filter is known as infinite-impulse-response because in general the impulse response is of infinite duration even when  $M$  and  $N$  are finite.

### 2.5.2 Finite-impulse-response (FIR) filters

An important special case of the IIR filter is the *finite-impulse-response* (FIR) filter, in which  $N$  is set to zero in the general filter equation:

$$y_n = \sum_{i=0}^M b_i x_{n-i}$$

This filter is unconditionally stable and has an impulse response equal to  $b_0, b_1, \dots, b_M, 0, 0, \dots$ . The transfer function is given by:

$$H(z) = B(z)$$

## 2.6 The discrete Fourier transform (DFT)

The Discrete Time Fourier Transform (DTFT) expresses the spectrum of a sampled signal in terms of the signal samples but is not computable on a digital machine for two reasons:

1. The frequency variable  $\omega$  is continuous.
2. The summation involves an infinite number of samples

The first problem may be overcome by simply choosing to evaluate the spectrum at a set of discrete frequencies. These may be arbitrarily chosen but it should be remembered that the spectrum is a periodic function of frequency (see figure 2.2) so that it is only necessary to consider frequencies in the range:

$$\omega T = 0 \rightarrow 2\pi$$

Although the frequencies may be arbitrarily chosen in this range, important computational advantages are to be gained from choosing  $N$  uniformly spaced frequencies, ie. at the frequencies:

$$\omega T = p \frac{2\pi}{N} \quad p \in \{0, 1, \dots, N-1\}$$

Thus equation (2.3) becomes:

$$G(e^{j\frac{2\pi}{N}p}) = \sum_{n=-\infty}^{\infty} g_n e^{-j\frac{2\pi}{N}np}$$

Although the above equation is a function of only discrete variables the problem remains of the infinite summation. If, however, the signal is non-zero only for samples  $g_0$  to  $g_{M-1}$  then the equation becomes:

$$G(e^{j\frac{2\pi}{N}p}) = \sum_{n=0}^{M-1} g_n e^{-j\frac{2\pi}{N}np} \quad (2.11)$$

In general, of course, the signal will not be of finite duration. However, if it is assumed that computing the transform of only  $M$  samples of the signal will lead to a reasonable estimate of the spectrum of the infinite duration signal then the above equation is computable in a finite time. (The implications of this assumption are investigated in the next section on data windowing.) Although the number of samples  $M$  is completely independent of the number of frequency points  $N$ , there is a considerable computational advantage to be gained from setting  $M = N$  and this will become clear later when the fast Fourier transform (FFT) is considered. Under this condition, equation (2.11) becomes:

$$G(e^{j\frac{2\pi}{N}p}) = \sum_{n=0}^{N-1} g_n e^{-j\frac{2\pi}{N}np} \quad (2.12)$$

It is often required to be able to compute the signal samples from the spectral values and this may be achieved by making use of the orthogonal properties of the discrete complex exponential as follows. Multiply each side of equation (2.12) by  $e^{j\frac{2\pi}{N}pq}$  and sum over  $p = 0$  to  $N - 1$ :

$$\begin{aligned} \sum_{p=0}^{N-1} G(e^{j\frac{2\pi}{N}p}) e^{j\frac{2\pi}{N}pq} &= \sum_{p=0}^{N-1} \sum_{n=0}^{N-1} g_n e^{-j\frac{2\pi}{N}np} e^{j\frac{2\pi}{N}pq} \\ &= \sum_{n=0}^{N-1} g_n \sum_{p=0}^{N-1} e^{j\frac{2\pi}{N}(q-n)p} \end{aligned}$$

Note that:

$$\sum_{p=0}^{N-1} e^{j\frac{2\pi}{N}(q-n)p} = \begin{cases} N & n = q \\ 0 & n \neq q \end{cases}$$

$$\therefore g_q = \frac{1}{N} \sum_{p=0}^{N-1} G(e^{j\frac{2\pi}{N}p}) e^{j\frac{2\pi}{N}pq} \quad (2.13)$$

Equations (2.12) and (2.13) are the discrete Fourier transform (DFT) pair, summarised as:

$$G(p) = \sum_{n=0}^{N-1} g_n e^{-j \frac{2\pi}{N} np} \quad (2.14)$$

*Discrete Fourier transform (DFT)*

$$g_n = \frac{1}{N} \sum_{p=0}^{N-1} G(p) e^{j \frac{2\pi}{N} pn} \quad (2.15)$$

*Inverse DFT*

## 2.7 Data windows

In the previous section the discrete Fourier transform (DFT) was derived for a signal  $\{g_n\}$  which is non-zero only for  $n \in \{0, 1, \dots, N-1\}$ . In most Fourier transform applications the signal will not be of finite duration. However we could force this condition by extracting a section of the signal with duration  $T_w$  and hoping that the spectrum of the finite duration signal would be a good approximation to the spectrum of the long duration signal. This can be analysed carefully and leads on to the topic of window design.

### 2.7.1 Continuous signals

It will be helpful firstly to consider continuous signals. Let  $g(t)$  be a continuous signal defined over all time with Fourier transform  $G(\omega)$ :

$$g(t) \Leftrightarrow G(\omega)$$

We wish to determine what relationship the spectrum of the windowed signal  $g_w(t)$ , shown in figure 2.4, has to the spectrum of  $g(t)$ , where  $g_w(t) = g(t) w(t)$

Transforming to the frequency domain and using the standard convolution result for the transform of multiplied signals, we obtain

$$G_w(\omega) = \int_{-\infty}^{\infty} g_w(t) e^{-j\omega t} dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} W(\lambda) G(\omega - \lambda) d\lambda \quad (2.16)$$

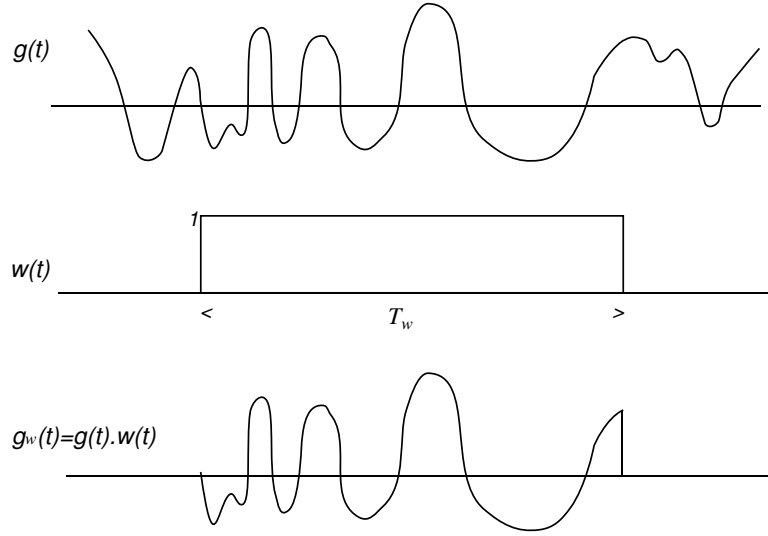


FIGURE 2.4. Windowing a continuous-time signal

The spectrum of the windowed signal is thus the convolution of the window's spectrum with the un-windowed signal's spectrum. The spectrum of the window is

$$\begin{aligned}
 W(\omega) &= \int_{-\infty}^{\infty} w(t) \cdot e^{-j\omega t} dt = \int_{-\frac{T_w}{2}}^{\frac{T_w}{2}} e^{-j\omega t} dt = \frac{1}{-j\omega} \left[ e^{-j\omega \frac{T_w}{2}} - e^{j\omega \frac{T_w}{2}} \right] \\
 &= \frac{2 \sin(\omega \frac{T_w}{2})}{\omega} = T_w \frac{\sin(\omega \frac{T_w}{2})}{(\omega \frac{T_w}{2})} \quad (2.17)
 \end{aligned}$$

as shown in figure 2.5. The effect of this convolution can be appreciated by considering the signal:

$$g(t) = 1 + e^{j\omega_o t}$$

ie. a d.c. offset added to a complex exponential with frequency  $\omega_o$ . The Fourier transform of this signal is:

$$G(\omega) = 2\pi\delta(\omega) + 2\pi\delta(\omega - \omega_o)$$

The spectrum of the windowed signal is then obtained as:

$$\begin{aligned}
 G_w(\omega) &= \int_{-\infty}^{\infty} \left\{ T_w \frac{\sin(\lambda \frac{T_w}{2})}{(\lambda \frac{T_w}{2})} \right\} \{ \delta(\omega - \lambda) + \delta(\omega - \lambda - \omega_o) \} d\lambda \\
 &= T_w \left\{ \frac{\sin(\omega \frac{T_w}{2})}{(\omega \frac{T_w}{2})} + \frac{\sin[(\omega - \omega_o) \frac{T_w}{2}]}{(\omega - \omega_o) \frac{T_w}{2}} \right\}
 \end{aligned}$$

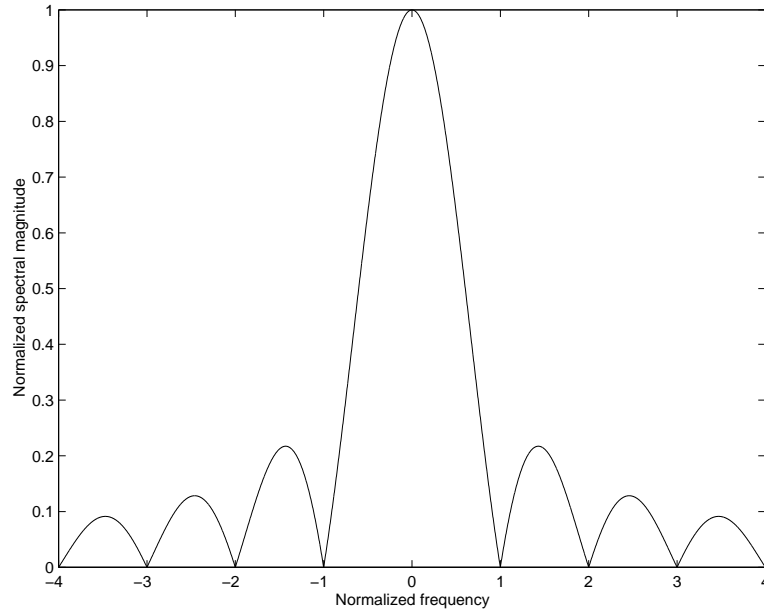


FIGURE 2.5. Spectral magnitude of rectangular window  $|W(\omega)|$  plotted as a function of normalised frequency  $\omega T_w/(2\pi)$

This result is plotted in figure 2.6 and we see that there are two effects.

1. Smearing. The discrete frequencies (in this case *d.c.* and  $\omega_0$ ) have become ‘smeared’ into a band of frequencies.
2. Spectral leakage. The component at *d.c.* ‘leaks’ into the  $\omega_0$  component as a result of the sidelobes of  $W(f)$ . This can be very troublesome when trying to measure the amplitude of a small component in the presence of a large component at a nearby frequency. This is illustrated for two complex exponentials with differing amplitudes in figure 2.7.

The two effects are not independent but roughly speaking the width of the lobes (or smearing) is an effect of the window duration but the sidelobes are due to the discontinuous nature of the window. A technique commonly used in spectral analysis is to employ a tapered window rather than the rectangular window. One such window is the ‘cosine arch’ or Hanning window, given by:

$$w(t) = \begin{cases} \frac{1}{2}[1 - \cos(\frac{2\pi t}{T_w})], & 0 \leq t \leq T_w \\ 0, & \text{otherwise} \end{cases}$$

and shown in figure 2.8 along with its spectrum, which has much reduced sidelobes but wider central lobe than the rectangular window.

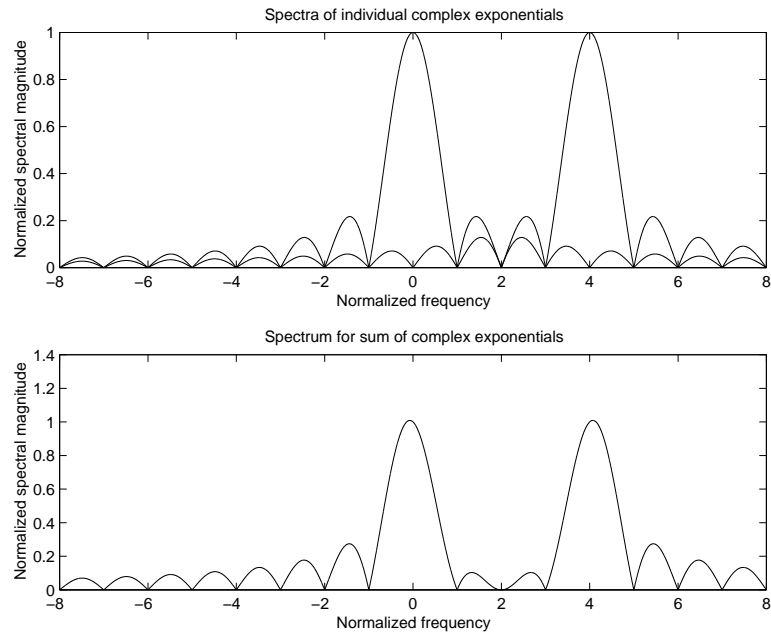


FIGURE 2.6. Top - spectra of individual d.c. component and exponential with normalised frequency of 4; bottom - overall spectrum  $|G_w(\omega)|$ .

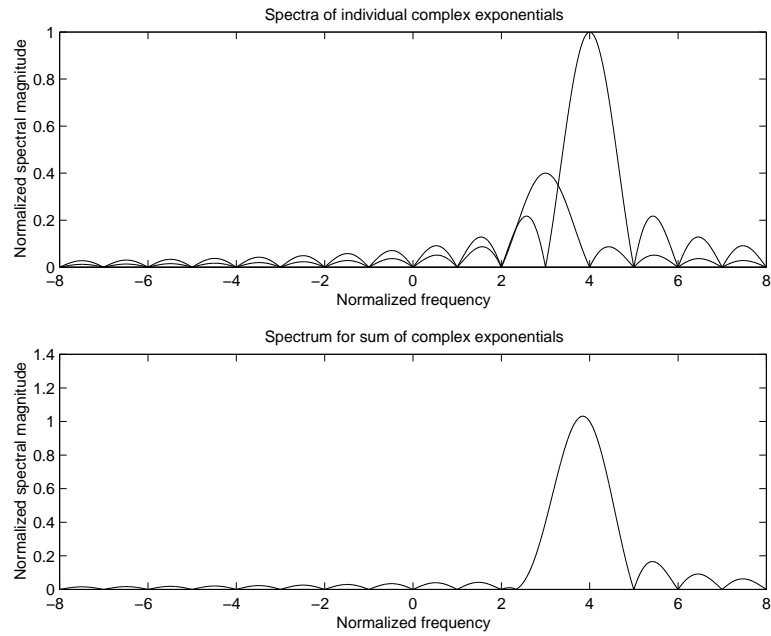


FIGURE 2.7. Top - spectra of two complex exponentials with normalised frequencies 3 and 4, first component is  $2/5$  the amplitude of second; bottom - spectrum of the sum, showing how the first component is completely lost owing to spectral leakage.



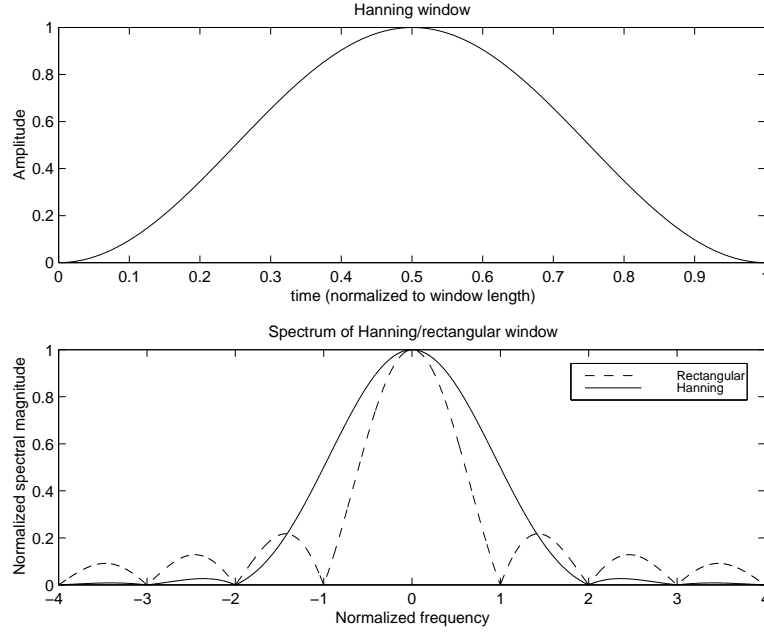


FIGURE 2.8. Hanning window (top) and its spectrum (bottom)

### 2.7.2 Discrete-time signals

Consider now the discrete case. We can go through the calculations for the windowed spectrum in a manner analogous to the continuous-time case:

$$\begin{aligned}
 G(e^{j\omega T}) &= \sum_{p=-\infty}^{\infty} g(pT) e^{-jp\omega T} \\
 G_w(e^{j\omega T}) &= \sum_{p=-\infty}^{\infty} \{g(pT) w(pT)\} e^{-jp\omega T} \\
 &= \sum_{p=-\infty}^{\infty} g(pT) \left\{ \frac{1}{2\pi} \int_0^{2\pi} W(e^{j\theta}) e^{jp\theta} d\theta \right\} e^{-jp\omega T} \\
 &= \frac{1}{2\pi} \int_0^{2\pi} W(e^{j\theta}) \sum_{p=-\infty}^{\infty} g(pT) e^{-jp(\omega T - \theta)} d\theta \\
 \therefore G_w(e^{j\omega T}) &= \frac{1}{2\pi} \int_0^{2\pi} W(e^{j\theta}) G(e^{j(\omega T - \theta)}) d\theta
 \end{aligned}$$

Once again, the spectrum of the windowed signal is the convolution of the infinite duration signal's spectrum and the window's spectrum. Note that all spectra in the discrete case are periodic functions of frequency.

As for the continuous case we can consider the use of tapered windows and one class of window functions is the generalised Hamming window given by

$$w(nT) = \alpha - (1 - \alpha) \cos\left(\frac{2\pi}{N}n\right) \quad n = 0, \dots, N - 1$$

A few specific values of  $\alpha$  are given their own names, as follows,

$\alpha = 1$	Rectangular window
$\alpha = 0.5$	Hanning window (Cosine Arch)
$\alpha = 0.54$	Hamming window

Figure 2.9 shows the window shapes and figure 2.10 the spectra (on a logarithmic scale) for several values of  $\alpha$ . Other commonly used data windows include the Kaiser, Bartlett and Parzen windows, each giving a different trade-off in side-lobe and central lobe properties. Choice of windows will depend upon the application, involving a suitable trade-off between side-lobe height and central lobe width. Other commonly used windows include the Bartlett, Tukey and Parzen windows.

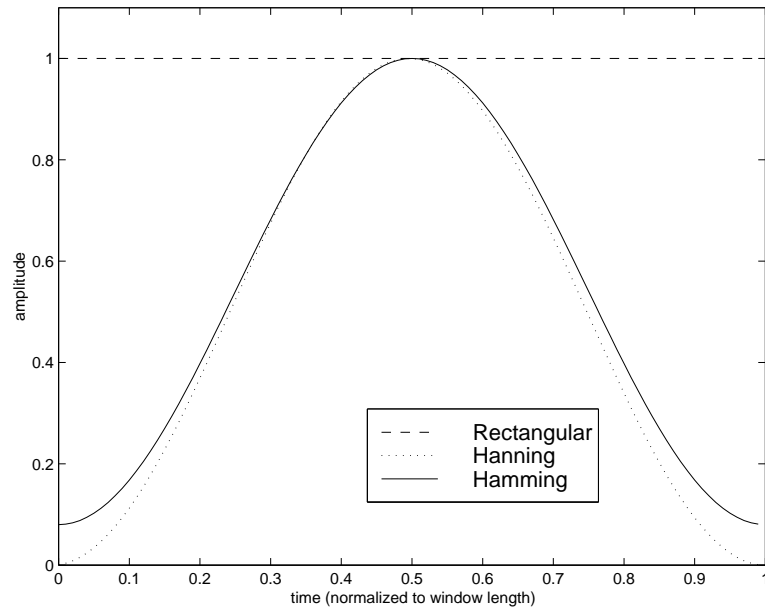


FIGURE 2.9. Discrete window functions

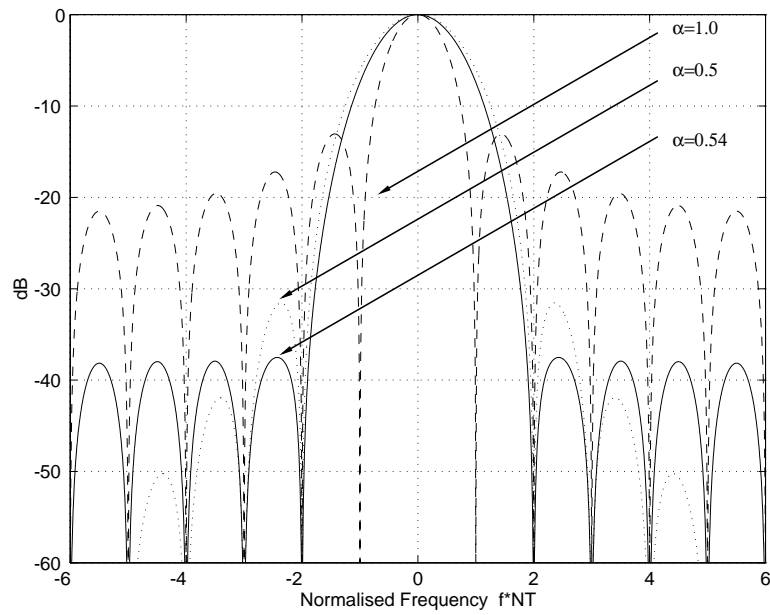


FIGURE 2.10. Discrete window spectra

## 2.8 The fast Fourier transform (FFT)

The Discrete Fourier transform (DFT) of a sequence of data  $\{x_n\}$  and its inverse is given by:

$$X(p) = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}np} \quad (2.18)$$

$$x_n = \frac{1}{N} \sum_{p=0}^{N-1} X(p) e^{j\frac{2\pi}{N}np} \quad (2.19)$$

for  $n \in \{0, N-1\}$  and  $p \in \{0, N-1\}$ .

Computation of  $X(p)$  for  $p = 0, 1, \dots, N-1$  requires on the order of  $N^2$  complex multiplications and additions. The Fast Fourier Transform algorithm reduces the required number of arithmetic operations to the order of  $\frac{N}{2} \log_2(N)$ , which provides a critical degree of speed-up for applications involving large  $N$ .

The first stage of the algorithm is to rewrite equation (2.18) in terms of the even-indexed and odd-indexed data  $x_n$ :

$$\begin{aligned} X(p) &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)p} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}np} + e^{-j\frac{2\pi}{N}p} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}np} \end{aligned} \quad (2.20)$$

It can be seen that computation of the length  $N$  DFT has been reduced to the computation of two length  $\frac{N}{2}$  DFTs and an additional  $N$  complex multiplications for the complex exponential outside the second summation. It would appear, at first sight, that it is necessary to evaluate equation (2.20) for  $p = 0, 1, \dots, N-1$ . However, this is not necessary as may be seen by considering the equation for  $p \geq (\frac{N}{2})$ :

$$\begin{aligned} X(p + N/2) &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}n(p + \frac{N}{2})} + e^{-j\frac{2\pi}{N}(p + \frac{N}{2})} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}n(p + \frac{N}{2})} \\ &= \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}np} - e^{-j\frac{2\pi}{N}p} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}np} \end{aligned} \quad (2.21)$$

Comparing equation (2.21) for  $X(p + \frac{N}{2})$  with equation (2.20) for  $X(p)$  it can be seen that the only difference is the sign between the two summations. Thus it is necessary to evaluate equation (2.20) only for  $p = 0, 1, \dots, \frac{N}{2} - 1$ , storing the results of the two summations separately for each  $p$ . The values of  $X(p)$  and  $X(p + \frac{N}{2})$  can then be evaluated as the sum and difference of the two summations as indicated by equations (2.20) and (2.21).

Thus the computational load for an  $N$ -point DFT has been reduced from  $N^2$  complex arithmetic operations to  $2(\frac{N}{2})^2 + \frac{N}{2}$ , which is approximately half the computation for large  $N$ . This is the first stage in the FFT derivation. The rest of the derivation proceeds by re-applying the same procedure a number of times.

We now define the notation:

$$W = e^{-j\frac{2\pi}{N}}$$

and define the FFT *butterfly* to be as shown in figure 2.11

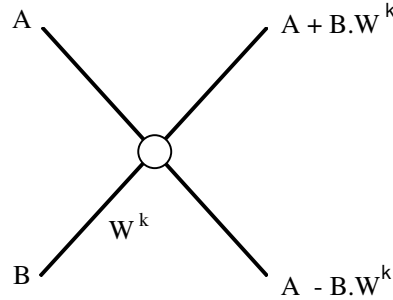


FIGURE 2.11. The FFT *butterfly*

The flow chart for incorporating this decomposition into the computation of an  $N = 8$  point DFT is shown in figure 2.12. Assuming that  $(\frac{N}{2})$  is even, the same process can be carried out on each of the  $(\frac{N}{2})$ -point DFTs to reduce the computation further. The flow chart for incorporating this extra stage of decomposition into the computation of the  $N = 8$  point DFT is shown in figure 2.13.

It can be seen that if  $N = 2^M$  then the process can be repeated  $M$  times to reduce the computation to that of evaluating  $N$  single point DFTs. Thus the flow chart for computing the  $N = 8$  point DFT is as shown in figure 2.14.

Examination of the final chart shows that it is necessary to shuffle the order of the input data. This data shuffle is usually termed *bit-reversal* for reasons that are clear if the indices of the shuffled data are written in binary.

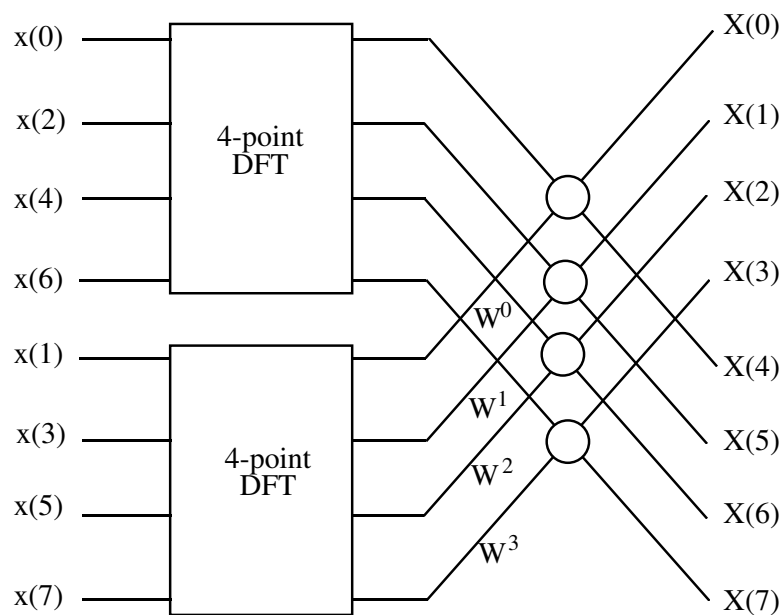


FIGURE 2.12. First Stage of FFT Decomposition

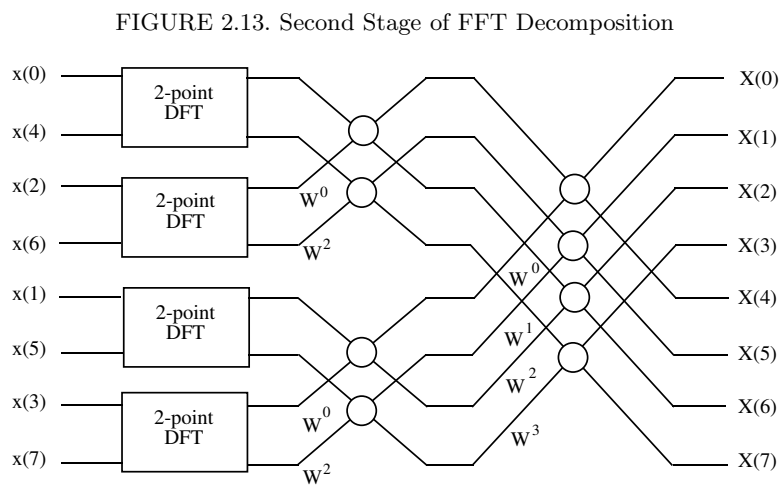
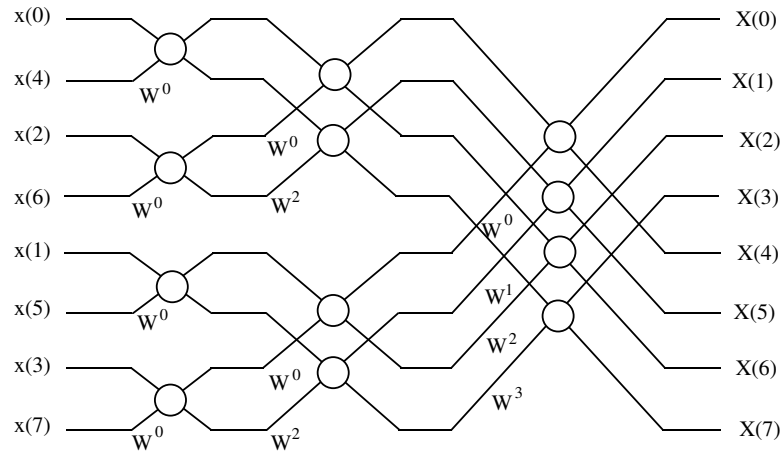


FIGURE 2.13. Second Stage of FFT Decomposition

FIGURE 2.14. Complete FFT Decomposition for  $N = 8$ 

Binary	Bit Reverse	Decimal
000	000	0
001	100	4
010	010	2
011	110	6
100	001	1
101	101	5
110	011	3
111	111	7

It has been shown that the process reduces, at each stage, the computation by half but introduces an extra  $\frac{N}{2}$  multiplications to account for the complex exponential term outside the second summation term in the reduction. Thus, for the condition  $N = 2^M$ , the process can be repeated  $M$  times to reduce the computation to that of evaluating  $N$  single point DFTs which requires no computation. However, at each of the  $M$  stages of reduction an extra  $\frac{N}{2}$  multiplications are introduced so that the total number of arithmetic operations required to evaluate an  $N$ -point DFT is  $\frac{N}{2} \log_2(N)$ .

The FFT algorithm has a further significant advantage over direct evaluation of the DFT expression in that computation can be performed *in-place*. This is best illustrated in the final flow chart where it can be seen that after two data values have been processed by the *butterfly* structure, those data are not required again in the computation and they may be replaced, in the

computer memory, with the values at the output of the *butterfly* structure.

## 2.9 Conclusion

In this chapter we have briefly described the discrete time signal theory upon which the rest of the book is based. The material thus far has considered only deterministic signals with known, fixed waveform. In the next two chapters we will introduce the idea of signals drawn at random from some large ensemble of possibilities according to the laws of probability.



# 3

## Probability Theory and Random Processes

In this chapter the basic results of probability theory and random processes are developed. These concepts are fundamental to most of the subsequent chapters since our methodology is generally based upon probabilistic arguments. It is recommended that this chapter is used as a reference rather than for learning about probability from scratch as it comprises mostly a list of results used later in the book without detailed discussion. The material covered is sufficient for the purposes of this text, but for further mathematical rigour and detail see for example Gray and Davisson [87], Papoulis [148] or Doob [48].

### 3.1 Random events and probability

Probability theory is used to give a mathematical description of the behaviour of real-world systems which involve **random events**. Such a system might be as simple as a coin-flipping experiment, in which we are interested in whether ‘Heads’ or ‘Tails’ is the outcome, or it might be more complex, as in the study of random errors in a coded digital datastream (e.g. a CD recording or a digital mobile phone). Here we summarise the main results and formalise the intuitive ideas of Events and the **Axioms of Probability** into a **Probability Space** which encapsulates everything we need to know about a system of random events.

The probability results given in this section are strictly event-based, since it is relatively easy to obtain results which have all the desired properties of

a probabilistic system. In later sections we will use these results as the basis for study of more advanced topics in random signal theory which involve random signals rather than individual events. We will see, however, that random signal theory can be formulated in terms of event-based results or as limiting cases of those results.

### 3.1.1 Frequency-based interpretation of probability

An intuitive interpretation of probability is as follows:

Suppose we observe  $M$  identical experiments whose outcome is a member of the set  $\Omega = \{\omega_1, \omega_2, \dots, \omega_N\}$ . The outcome is  $\omega_i$  in  $M_i$  of the  $M$  experiments. The *Probability* of  $\omega_i$  is the proportion of experiments in which  $\omega_i$  is the observed in the limit as the total number of experiments tends to infinity:

$$\Pr\{\omega_i\} = \lim_{M \rightarrow \infty} \frac{M_i}{M}$$

## 3.2 Probability spaces

The term *Random Experiment* is used to describe any situation which has a set of possible outcomes, each of which occurs with a particular *Probability*.

Any random experiment can be described completely by its *Probability Space*,  $(\Omega, \mathcal{F}, P)$ :

1. **Sample Space**  $\Omega = \{\omega_1, \omega_2, \dots, \omega_N\}$ , the set of possible ‘elementary’ outcomes of the experiment.
2. **Event Space**  $\mathcal{F}$ : the space of events  $F$  for which a probability is assigned, including  $\emptyset$  (the empty set) and  $\Omega$ . Events are collections of elementary outcomes. New events can be obtained by set-theoretic operations on other events, e.g.

$$H = F \cup G, H = F \cap G, H = F^c$$

For our purposes, consider that  $\mathcal{F}$  contains all ‘physically meaningful’ events which can be obtained by set-theoretic operations on the sample space.<sup>1</sup>

---

<sup>1</sup>There is a great deal of mathematics behind the generation of appropriate event spaces, which are sometimes known as a *sigma field* or *sigma algebra*. In particular, for continuous sample spaces it is possible (but difficult!) to generate events for which a consistent system of probability *cannot* be devised. However, such events are not physically meaningful and we can ignore them.

3. **Probability Measure  $P$ :** A function, literally a ‘measure of probability’, defining the probability  $P(F)$  of each member of the event space. Interpret  $P$  as:

$$P(F) = \Pr\{\text{‘The outcome is in } F\text{’}\} = \Pr\{\omega \in F\}$$

$P$  must obey the axioms of probability:

- (a)  $0 \leq P(F) \leq 1$ . The probability of any event must lie between 0 and 1.
- (b)  $P(\Omega) = 1$ . The probability of the outcome lying within the sample space is 1 (i.e. certain).
- (c)  $P(F \cup G) = P(F) + P(G)$  for events  $F$  and  $G$  s.t.  $F \cap G = \emptyset$  (‘mutually exclusive’). This means that for events  $F$  and  $G$  in  $\Omega$  with no elements in common the probability of  $F$  OR  $G$  occurring is the sum of their individual probabilities.

*Example.* A simple example is the coin-flipping experiment for which we can easily write down a probability space:

$$\begin{aligned}\Omega &= \{H, T\} \\ \mathcal{F} &= \{\{H\}, \{T\}, \Omega, \emptyset\} \\ P(\{H\}) &= \frac{1}{2}, \quad P(\{T\}) = \frac{1}{2}, \quad P(\Omega) = 1, \quad P(\emptyset) = 0\end{aligned}$$

where  $\emptyset$  denotes the empty set, i.e. no outcome whatsoever. The event space  $\mathcal{F}$  is chosen to contain all the events we might wish to consider: the probability of heads ( $P(H) = 1/2$ ), the probability of tails ( $P(T) = 1/2$ ), the probability of heads OR tails ( $P(\Omega = \{H, T\}) = 1$ , the certain event), the probability of no outcome at all ( $P(\emptyset) = 0$ ).

The above is an example of a *discrete* probability space, in which the sample space can be mapped with a one-to-one equivalence onto a subset of the integers  $\mathcal{Z}$ .<sup>2</sup> In much of the ensuing work, however, we will be concerned with *continuous* sample spaces which cannot be mapped onto the integers in this way:

*Example.* Consider a random thermal noise voltage  $V$  measured across a resistor. The sample space  $\Omega$  would in this case be the real line:

$$\Omega = \Re = (-\infty, +\infty)$$

which cannot be mapped onto the integers. The event space is now much harder to construct mathematically. However, we can answer all ‘physically

---

<sup>2</sup>The number of elements in the sample space is then referred to as *countable*.

meaningful' questions such as "what is the probability that the voltage lies between -1 and +1" by including in the event space all<sup>3</sup> subsets  $F$  of the real line.

The event probabilities can then be calculated by integrals of the probability density function (PDF)  $f(v)$  over the appropriate subset  $F$  of the real line (see section on random variables):

$$P(F) = \Pr\{V \in F\} = \int_{v \in F} f(v) dv$$

e.g. for  $F = (a, b]$  we have

$$P(F) = \Pr\{a < V \leq b\} = \int_{v=a^+}^b f(v) dv$$

The notion of a probability space formalises our description of a random experiment. In the examples given in this book we will not lay out problems in such a formal manner. Given a particular experiment with a given sample space it will usually be possible to pose and solve problems without explicitly constructing an event space. However, it is useful to remember that a Probability Space  $(\Omega, \mathcal{F}, P)$  can be devised which fully describes the properties of any random system we are likely to encounter.

### 3.3 Fundamental results of event-based probability

#### 3.3.1 Conditional probability

Consider events  $F$  and  $G$  in some probability space. Conditional probability, denoted  $P(F|G)$ , is defined as the probability that event  $F$  occurs *given* the prior knowledge that event  $G$  has occurred. A formula for calculating conditional probability can be obtained by reasoning from the axioms of probability as:

$$\boxed{P(F|G) = \frac{P(F \cap G)}{P(G)}, \quad P(G) > 0} \quad (3.1)$$

*Conditional Probability*

---

<sup>3</sup>with the exception of the difficult cases mentioned in a previous footnote, which we ignore as before

Note that conditional probability is undefined when  $P(G) = 0$ , since event  $F|G$  is meaningless if  $G$  is impossible anyway! The event  $F \cap G$  has the interpretation ‘the outcome is in  $F$  AND  $G$ ’. Some readers may be more familiar with the notation  $(F, G)$  which means the same thing.

### 3.3.2 Bayes rule

The equivalent formulation for  $G$  conditional upon  $F$  is clearly  $P(G|F) = \frac{P(G \cap F)}{P(F)}$ . Since  $P(F \cap G) = P(G \cap F)$  we can substitute the resulting expression for  $P(G \cap F)$  in place of  $P(F \cap G)$  in result (3.1) to obtain *Bayes Rule*:

$$P(F|G) = \frac{P(G|F)P(F)}{P(G)}$$

*Bayes Rule*

This equation is fundamental to many inferential procedures, since it tells us how to proceed from a probabilistic description of an event  $G$ , which has been directly observed, to some unobserved event  $F$  whose probability we need to know. Within an inferential framework,  $P(F)$  is known as the prior or *a priori* probability since it reflects prior knowledge about event  $F$  before event  $G$  has been observed;  $P(F|G)$  is known as the posterior or *a posteriori* probability since it gives the probability of  $F$  after event  $G$  has been observed;  $P(G|F)$  is known as the likelihood and  $P(G)$  as the ‘total’ or ‘marginal’ probability for  $G$ .

### 3.3.3 Total probability

If the total probability  $P(G)$  is unknown for a particular problem it can be calculated by forming a *partition*  $\{F_1, F_2, \dots, F_n\}$  of the sample space  $\Omega$ . By this we mean that we find a set of events from the event space  $\mathcal{F}$  which are mutually exclusive (i.e. there is no ‘overlap’ between any two members of the partition) and whose union is the complete sample space. For example,  $\{1, 2, 3, 4, 5, 6\}$  forms a partition in the die-throwing experiment and  $\{H, T\}$  forms a partition of the coin-flipping experiment. The total probability is then given by

$$P(G) = \sum_{i=1}^n P(G|F_i)P(F_i) \quad (3.2)$$

*Total probability*

### 3.3.4 Independence

If one event has no effect upon the outcome of another event, and *vice versa*, then the two events are termed *independent*. Mathematically this is expressed in terms of the joint probability  $P(F \cap G)$  for the events, i.e. the probability that the outcome is in both  $F$  and  $G$ . The joint probability is expressed in the way one might expect, as the product of the individual event probabilities:

$$P(F \cap G) = P(F)P(G) \quad \Longleftrightarrow \quad F \text{ and } G \text{ independent}$$

*Independent events*

Note that this formula, which forms the *definition* of independent events, is a special case of the conditional probability expression (3.1) in which  $P(F|G) = P(F)$ .

## 3.4 Random variables

A random variable is intuitively just a numerical quantity which takes on a random value. However, in order to derive the properties of random variables they are defined in terms of an underlying probability space. Consider a probability space  $(\Omega, \mathcal{F}, P)$ . The sample and event spaces for this probability space are in general defined in terms of abstract quantities such as ‘Heads’ and ‘Tails’ in the coin-tossing experiment, or ‘On’ and ‘Off’ in a logic circuit. As seen in the last section, the laws of probability can be applied directly in this abstract domain. However, in many cases we will wish to study the properties of a numerical outcome such as a random *voltage* or *current* rather than these abstract quantities. A mapping  $X(\omega)$  which assigns a real numerical value to each outcome  $\omega \in \Omega$  is termed a **Random Variable**. The theory of random variables will form the basis of much of the work in this book. We consider only real-valued random variables, since it is straightforward to extend the results to complex-valued variables.

### 3.5 Definition: random variable

Given a probability space  $(\Omega, \mathcal{F}, P)$ , a random variable is a function  $X(\omega)$  which maps each element  $\omega$  of the sample space  $\Omega$  onto a point on the real line. We will usually refer to the mapping  $X(\omega)$  and the underlying probability space simply as ‘random variable (or RV)  $X$ ’.

#### 3.5.0.0.1 Example: Discrete random variable

If the mapping  $X(\omega)$  can take on only a *countable* number of values on the real line, the random variable is termed *discrete*.

For a logic gate with states ‘On’ and ‘Off’ define the random variable:

$$X(\text{On}) = 1, \quad X(\text{Off}) = 0$$

However, if one wished to study the properties of a line transmission system which transmits an ‘On’ as +1.5V and an ‘Off’ as -1.5V, then a more natural choice of RV would be:

$$X(\text{On}) = +1.5, \quad X(\text{Off}) = -1.5$$

#### 3.5.0.0.2 Example: Continuous random variable

The random noise voltage  $V$  measured across a resistor  $R$  is a *continuous* random variable. Since the sample space  $\Omega$  is the measured voltage itself, i.e.  $V(\omega) = \omega$ , the random variable  $V$  is termed ‘directly given’. Alternatively, we may wish to consider the instantaneous power  $V^2/R$  across the resistor, in which case we can define a RV  $W(\omega) = \omega^2/R$ .

### 3.6 The probability distribution of a random variable

The distribution  $P_X$  of a RV  $X$  is simply a probability measure which assigns probabilities to events on the real line. The distribution  $P_X$  answers questions of the form ‘what is the probability that  $X$  lies in subset  $F$  of the real line?’.  $P_X$  is obtained via the ‘inverse mapping’ method:

$$P_X(F) = \Pr\{X \in F\} = P(X^{-1}(F)) \quad (3.3)$$

where the inverse mapping is defined mathematically as:

$$X^{-1}(F) = \{\omega : X(\omega) \in F\}, \quad (F \subset \mathbb{R}) \quad (3.4)$$

In practice we will summarise  $P_X$  by its *Probability Mass Function* (PMF, discrete variables only), *Cumulative Distribution Function* (CDF) or *Probability Density Function* (PDF)

### 3.6.1 Probability mass function (PMF) (discrete RVs)

Suppose the variable can take a set of  $M$  real values  $\{x_1, \dots, x_i, \dots, x_M\}$ . Then the probability mass function (PMF) is defined as:

$$p_X(x_i) = \Pr\{X = x_i\} = P_X(\{x_i\}), \quad \sum_{i=1}^M p_X(x_i) = 1$$

*Probability mass function (PMF)*

### 3.6.2 Cumulative distribution function (CDF)

The cumulative distribution function (CDF) can be used to describe discrete, continuous or mixed discrete/continuous distributions:

$$F_X(x) = \Pr\{X \leq x\} = P_X((-\infty, x])$$

*Cumulative distribution function (CDF)*

The following properties follow directly from the Axioms of Probability:

1.  $0 \leq F_X(x) \leq 1$
2.  $F_X(-\infty) = 0, \quad F_X(\infty) = 1$
3.  $F_X(x)$  is non-decreasing as  $x$  increases
4.  $\Pr\{x_1 < X \leq x_2\} = F_X(x_2) - F_X(x_1)$

Note also that  $\Pr\{X > x\} = 1 - F_X(x)$ .

Where there is no ambiguity we will usually drop the subscript ' $X$ ' and refer to the CDF as  $F(x)$ .

### 3.6.3 Probability density function (PDF)

The probability density function (PDF) is defined as the derivative of the CDF with respect to  $x$ :

$$f_X(x) = \frac{\partial F_X(x)}{\partial x}$$



*Probability density function (PDF)*

Again, its properties follow from the axioms of probability:

1.  $f_X(x) \geq 0$
2.  $\int_{-\infty}^{\infty} f_X(x) dx = 1$
3.  $F_X(x) = \int_{-\infty}^x f_X(\alpha) d\alpha$
4.  $\int_{x_1}^{x_2} f_X(\alpha) d\alpha = \Pr\{x_1 < X \leq x_2\}$

As for the CDF we will often drop the subscript and refer simply to  $f(x)$  when no confusion can arise.

The PDF can be viewed as a continuous *frequency histogram* for the RV since it represents the relative frequency with which a particular value is observed in many trials under the same conditions.

From property 4 we can obtain the following important result for the probability that  $X$  lies within a slice of infinitesimal width  $\delta x$ :

$$\begin{aligned} \lim_{\delta x \rightarrow 0} \Pr\{x < X \leq x + \delta x\} &= \lim_{\delta x \rightarrow 0} \int_x^{x+\delta x} f_X(\alpha) d\alpha \\ &= f_X(x) \delta x \end{aligned}$$

Note that this probability tends to zero as  $\delta x$  goes to zero, provided that  $f_X$  contains no delta functions.

For purely discrete RVs:

$$f_X(x) = \sum_{i=1}^M p_i \delta(x - x_i)$$

where the  $p_i$ 's are the PMF 'weights' for the discrete random variable.

### 3.7 Conditional distributions and Bayes rule

Conditional distributions and Bayes Rule are fundamental to inference in general and in particular to the analysis of complex random systems. Many technical questions will be of the form 'what can I infer about parameter  $a$  in a system given that I have made the related observation  $b$ ?'. Such questions can be answered using conditional probability.

### 3.7.1 Conditional PMF - discrete RVs

The results for PMFs apply for discrete random variables and are defined only at the values:

$$x \in \{x_1, \dots, x_M\}, \quad y \in \{y_1, \dots, y_N\}$$

Since ‘ $X = x_i$ ’ is an event with non-zero probability for a discrete random variable, we obtain conditional probability and related results directly from event-based probability, conditioning upon either some arbitrary event  $G$  or the value of a second random variable  $y$ :

$$p(x|G) = \Pr\{ (X = x) | G \} = \frac{P((X = x) \text{ AND } G)}{P(G)}$$

(Conditional upon arbitrary event  $G$ )

$$p(x|y) = \Pr\{ (X = x) | (Y = y) \} = \frac{p(x, y)}{p(y)}$$

(Conditional upon a second discrete RV  $Y = y$ )

Bayes rule and total probability for PMFs are then obtained directly as:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

*Bayes Rule (PMF)*

$$p(y) = \sum_{i=1}^M p(y|x_i)p(x_i)$$

*Total Probability (PMF)*

Note that  $p(x, y)$  is the *joint* PMF:

$$p(x, y) = \Pr\{ (X = x) \text{ AND } (Y = y) \}$$

Conditional and joint PMFs are positive and obey the same normalisation rules as ordinary PMFs:

$$\sum_{i=1}^M p(x_i|y) = 1, \quad \sum_{i=1}^M \sum_{j=1}^N p(x_i, y_j) = 1$$

### 3.7.2 Conditional CDF

Apply the same reasoning using events of the form ' $X \leq x$ ' and ' $Y \leq y$ ':

$$F(x|y) = \Pr\{(X \leq x)|(Y \leq y)\} = \frac{F(x, y)}{F(y)}$$

This leads directly to Bayes rule for CDFs:

$$F(x|y) = \frac{F(y|x)F(x)}{F(y)}$$

*Bayes Rule (CDF)*

$F(x|y)$  is the *conditional* CDF and  $F(x, y)$  is the *joint* CDF:

$$F(x, y) = \Pr\{(X \leq x) \text{ AND } (Y \leq y)\}$$

Conditional CDFs have similar properties to standard CDFs, i.e.  $F_{X|Y}(-\infty|y) = 0$ ,  $F_{X|Y}(+\infty|y) = 1$ , etc.

### 3.7.3 Conditional PDF

The conditional PDF is obtained as the derivative of the conditional CDF:

$$f(x|G) = \frac{\partial F(x|G)}{\partial x}$$

It has similar properties and interpretation as the standard PDF (i.e.  $f(x|G) > 0$ ,  $\int f(x|G) dx = 1$ ,  $P(X \in A|G) = \int_A f(x|G) dx$ ).

However, since the event ' $X = x$ ' has zero probability for continuous random variables, the PDF conditional upon  $X = x$  is not directly defined. We can obtain the required result as a limiting case:

$$\begin{aligned} \Pr\{G|X = x\} &= \lim_{\delta x \rightarrow 0} \Pr\{G|x < X \leq x + \delta x\} \\ &= \lim_{\delta x \rightarrow 0} \frac{\Pr\{x < X \leq x + \delta x|G\}P(G)}{\Pr\{x < X \leq x + \delta x\}} \\ &= \lim_{\delta x \rightarrow 0} \frac{(F(x + \delta x|G) - F(x|G))P(G)}{f(x)\delta x} \\ &= \lim_{\delta x \rightarrow 0} \left( \frac{(F(x + \delta x|G) - F(x|G))}{\delta x} \right) \frac{P(G)}{f(x)} \\ &= \frac{f(x|G)P(G)}{f(x)} \end{aligned}$$

Now taking  $G$  as the event  $Y \leq y$ :

$$\begin{aligned}
 F(y|X=x) &= \Pr\{Y \leq y|X=x\} \\
 &= \frac{f(x|Y \leq y)F(y)}{f(x)} \\
 &= \frac{\frac{\partial}{\partial x}(F(x|y)) F(y)}{f(x)} \\
 &= \frac{\frac{\partial}{\partial x}(F(x,y))}{f(x)}
 \end{aligned}$$

from whence:

$$\begin{aligned}
 f(y|x) &= \frac{\partial F(y|X=x)}{\partial y} \\
 &= \frac{\frac{\partial^2}{\partial x \partial y} F(x,y)}{f(x)} \\
 &= \frac{f(x,y)}{f(x)}
 \end{aligned}$$

where  $f(x,y)$  is the joint PDF, defined as:

$$f(x,y) = \frac{\partial^2 F(x,y)}{\partial x \partial y}$$

*Joint PDF*

The conditional probability rule for PDFs leads as before to Bayes Rule:

$$f(x|y) = \frac{f(y|x) f(x)}{f(y)}$$

*Bayes rule (PDF)*

Total probability is obtained as an integral:

$$\begin{aligned}
 \int f(y|x) f(x) dx &= \int \frac{f(x|y) f(y)}{f(x)} f(x) dx \\
 &= \left( \int f(x|y) dx \right) f(y) \\
 &= f(y)
 \end{aligned}$$

i.e. the total probability  $f(y)$  is given by:

$$f(y) = \int f(y|x)f(x) dx = \int f(y, x) dx$$

*Total Probability (PDF)*

This important result is often referred to as the *Marginalisation Integral* and  $f(y)$  as the *Marginal Probability*.

### 3.7.3.0.3 Example: the sum of 2 random variables

Consider the random variable  $Y$  formed as the sum of two independent random variables  $X_1$  and  $X_2$ :

$$Y = X_1 + X_2$$

$X_1$  has PDF  $f_1(x_1)$  and  $X_2$  has PDF  $f_2(x_2)$ .

We can write the joint PDF for  $y$  and  $x_1$  by rewriting the conditional probability formula:

$$f(y, x_1) = f(y|x_1) f_1(x_1)$$

It is clear that the event ‘ $Y$  takes the value  $y$  conditional upon  $X_1 = x_1$ ’ is equivalent to  $X_2$  taking a value  $y - x_1$  (since  $X_2 = Y - X_1$ ). Hence  $f(y|x_1) = f_{X_2|X_1}(y - x_1|x_1) = f_2(y - x_1)$ <sup>4</sup> and  $f(y)$  is obtained as follows:

$$\begin{aligned} f(y) &= \int f(y, x_1) dx_1 \\ &= \int f(y|x_1) f_1(x_1) dx_1 \\ &= \int f_2(y - x_1) f_1(x_1) dx_1 \\ &= f_1 * f_2 \end{aligned}$$

where ‘ $*$ ’ denotes the convolution operation. In other words, when two independent random variables are added, the PDF of their sum equals the convolution of the PDFs for the two original random variables.

---

<sup>4</sup>This ‘change of variables’ can be proved using results in a later section on functions of random variables

### 3.8 Expectation

Expectations form a fundamental part of random signal theory. Examples of expectations are the mean and standard deviations of a random variable, although the concept is much more general than this.

The mean of a random variable  $X$  is defined as:

$$E[X] = \overline{X} = \int_{x=-\infty}^{+\infty} x f_X(x) dx$$

*Mean value of  $X$*

If a second RV  $Y$  is defined to be a function  $Y = g(X)$  then the expectation of  $Y$  is obtained as:

$$E[Y] = \int_{x=-\infty}^{+\infty} g(x) f_X(x) dx$$

*Expectation of  $Y = g(X)$*

Expectation is a *linear operator*:

$$E[a g_1(X) + b g_2(X)] = a E[g_1(X)] + b E[g_2(X)]$$

Conditional expectations are defined in a similar way:

$$E[X|G] = \int x f_{X|G}(x|G) dx, \quad E[Y|G] = \int g(x) f_{X|G}(x|G) dx$$

*Conditional expectation*

Some important examples of expectation are as follows:

$$E[X^n] = \int x^n f_X(x) dx$$

*n*th order moment

$$E[(X - \bar{X})^n] = \int (x - \bar{X})^n f_X(x) dx$$

Central moment

The central moment with  $n = 2$  gives the familiar variance of a random variable:

$$E[(X - \bar{X})^2] = \int (x - \bar{X})^2 f_X(x) dx = E[X^2] - (\bar{X})^2$$

Variance

### 3.8.1 Characteristic functions

Another important example of expectation is the characteristic function  $\Phi_X(\omega)$  of a random variable  $X$ :

$$\begin{aligned} \Phi_X(\omega) &= E[e^{j\omega X}] \\ &= \int_{-\infty}^{+\infty} e^{j\omega x} f_X(x) dx \end{aligned}$$

which is equivalent to the Fourier transform of the PDF evaluated at  $\omega' = -\omega$ . Some of the properties of the Fourier Transform can be applied usefully to the characteristic function, in particular:

1. **Convolution - (sums of independent RVs)** We know that for sums of random variables the distribution is obtained by convolving the distributions of the component parts:

$$Y = \sum_{i=1}^N X_i$$

and

$$\begin{aligned} f_Y &= f_{X_1} * f_{X_2} \dots * f_{X_N} \\ \implies \Phi_Y(\omega) &= \prod_{i=1}^N \Phi_{X_i}(\omega) \end{aligned}$$

This result allows the calculation of the characteristic function for sums of independent random variables as the product of the component characteristic functions.

2. **Inversion** In order to invert back from the characteristic function to the PDF we apply the inverse Fourier transform with frequency negated:

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-j\omega x} \Phi_X(\omega) d\omega$$

### 3. Moments

$$\begin{aligned} \frac{\partial^n \Phi_X(\omega)}{\partial \omega^n} &= \int_{-\infty}^{+\infty} (jx)^n e^{j\omega x} f_X(x) dx \\ \implies E[X^n] &= 1/j^n \left. \frac{\partial^n \Phi_X(\omega)}{\partial \omega^n} \right|_{\omega=0} \end{aligned}$$

This result shows how to calculate moments of a random variable from its characteristic function.

## 3.9 Functions of random variables

Consider a random variable  $Y$  which is generated as a function  $Y = g(X)$ , where  $X$  has a known distribution  $P_X(x)$ . We can obtain the distribution for  $Y$ , the transformed random variable, directly by the ‘inverse image method’:



$$F_Y(y) = \Pr\{Y \leq y\} = P_X(\{x : g(x) \leq y\})$$

*Derived CDF for  $Y = g(X)$*

### 3.9.1 Differentiable mappings

If the mapping  $g(X)$  is differentiable then a simple formula can be obtained for the derived distribution of  $Y$ . Suppose that the equation  $y = g(x)$  has  $M$  solutions for a particular value of  $x$ :

$$g(x_1) = y, g(x_2) = y, \dots, g(x_M) = y$$

Defining  $g'(x) = \frac{\partial g(x)}{\partial x}$ , it can be shown that the derived PDF of  $Y$  is given by:

$$f_Y(y) = \sum_{i=1}^M \frac{f_X(x)}{|g'(x)|} \Big|_{x=x_i}$$

*PDF of  $Y = g(X)$*

#### 3.9.1.0.4 Example: $Y = X^2$ .

Suppose that  $X$  is a Gaussian random variable with mean zero:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (3.5)$$

The solutions of the equation  $y = x^2$  are

$$x_1 = +y^{1/2}, \quad x_2 = -y^{1/2}$$

and the modulus of the function derivative is

$$|g'(x)| = 2|x| = 2y^{1/2}$$

Substituting these into the formula for  $f_Y(y)$ , we obtain:

$$\begin{aligned} f_Y(y) &= \sum_{i=1}^2 \frac{f_X(x)}{|g'(x)|} \Big|_{x=x_i} \\ &= \frac{f_X(y^{1/2})}{2y^{1/2}} + \frac{f_X(-y^{1/2})}{2y^{1/2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2 y}} e^{-\frac{y}{2\sigma^2}} \end{aligned}$$

which is the Chi-squared density with one degree of freedom.

### 3.10 Random vectors

A *Vector random variable* can be thought of simply as a collection of single random variables which will in general be interrelated statistically. As for single random variables it is conceptually useful to retain the idea of a mapping from an underlying Probability Space  $(\Omega, \mathcal{F}, P)$  onto a real-valued random vector:

$$\mathbf{X}(\omega) = \begin{bmatrix} X_1(\omega) \\ X_2(\omega) \\ \vdots \\ X_N(\omega) \end{bmatrix}, \quad \omega \in \Omega, \quad \mathbf{X}(\omega) \in \mathbb{R}^N$$

i.e. the mapping is from the (possibly highly complex) sample space  $\Omega$  onto  $\mathbb{R}^N$ , the space of  $N$ -dimensional real vectors.

A vector cumulative distribution function can be defined by extension of the single random variable definition:

$$\boxed{F_{\mathbf{X}}(x_1, x_2, \dots, x_N) = F_{\mathbf{X}}(\mathbf{x}) = \Pr\{X_1 \leq x_1, X_2 \leq x_2, \dots, X_N \leq x_N\}}$$

*Vector (joint) cumulative distribution function (CDF)*

i.e.  $F_{\mathbf{X}}(\mathbf{x})$  is the probability that  $X_1 \leq x_1$  AND  $X_2 \leq x_2$  AND  $\dots$   $X_N \leq x_N$ . The vector CDF has the following properties:

1.  $0 \leq F_{\mathbf{X}}(\mathbf{x}) \leq 1$
2.  $F_{\mathbf{X}}(x_1, \dots, -\infty, \dots, x_N) = 0, \quad F_{\mathbf{X}}(\infty, \dots, \infty, \dots, \infty) = 1$

3.  $F_{\mathbf{X}}(\mathbf{x})$  is non-decreasing as any element  $x_i$  increases
4. Total Probability - e.g. for  $N = 4$ :

$$\begin{aligned} F(x_1, x_3) &= \Pr\{X_1 \leq x_1, X_2 \leq +\infty, X_3 \leq x_3, X_4 \leq +\infty\} \\ &= F(x_1, +\infty, x_3, +\infty) \end{aligned}$$

In a similar fashion the vector PDF is defined as:

$$f_{\mathbf{X}}(x_1 x_2 \dots x_N) = f_{\mathbf{X}}(\mathbf{x}) = \frac{\partial^N F_{\mathbf{X}}(\mathbf{x})}{\partial x_1 \partial x_2 \dots \partial x_N}$$

*Vector (joint) probability density function (PDF)*

The vector PDF has the following properties:

1.  $f_{\mathbf{X}}(\mathbf{x}) \geq 0$
2.  $\int_{x_1=-\infty}^{+\infty} \dots \int_{x_N=-\infty}^{+\infty} f_{\mathbf{X}}(\mathbf{x}) dx_1 dx_2 \dots dx_N = \int_{\mathbf{x}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = 1$
3.  $\Pr\{\mathbf{X} \in \mathcal{V}\} = \int_{\mathcal{V}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$
4. Expectation:

$$E[\mathbf{x}] = \int_{\mathbf{x}} \mathbf{x} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \quad E[g(\mathbf{x})] = \int_{\mathbf{x}} g(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

5. Probability of small volumes:

$$\begin{aligned} \Pr\{x_1 < X_1 \leq x_1 + \delta x_1, \dots, x_N < X_N \leq x_N + \delta x_N\} \\ \approx f_{\mathbf{X}}(x_1, x_2, \dots, x_N) \delta x_1 \delta x_2 \dots \delta x_N \end{aligned}$$

for small  $\delta x_i$ .

6. Marginalisation - integrate over unwanted components, e.g.

$$f(x_1, x_3) = \int_{x_2=-\infty}^{\infty} \int_{x_4=-\infty}^{\infty} f(x_1, x_2, x_3, x_4) dx_2 dx_4$$

### 3.11 Conditional densities and Bayes rule

These again are obtained as a direct generalisation of the results for single random variables. We treat only the case of PDFs for continuous random vectors as the discrete and mixed cases follow exactly the same form.

$$f_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \frac{f_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})}{f_{\mathbf{Y}}(\mathbf{y})}$$

*Conditional probability (random vector)*

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

*Bayes rule (random vector)*

### 3.12 Functions of random vectors

Consider a vector function of a vector random variable:

$$\mathbf{Y} = \begin{bmatrix} g_1(\mathbf{X}) \\ g_2(\mathbf{X}) \\ \vdots \\ g_N(\mathbf{X}) \end{bmatrix} = \mathbf{g}(\mathbf{X}), \quad \mathbf{g}(\mathbf{X}) \in \mathbb{R}^N$$

What is the derived PDF  $f_{\mathbf{Y}}(\mathbf{y})$  given  $f_{\mathbf{X}}(\mathbf{x})$  and  $\mathbf{g}()$ ?

Assuming that  $\mathbf{g}()$  is one-to-one, invertible and differentiable, the derived PDF is obtained as:

$$f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{|\det(\mathbf{J})|} f_{\mathbf{X}}(\mathbf{x})|_{\mathbf{x}=\mathbf{g}^{-1}(\mathbf{y})}$$

where the Jacobian  $|\det(\mathbf{J})|$  gives the ratio between the infinitesimal  $N$ -dimensional volumes in the  $\mathbf{x}$  space and  $\mathbf{y}$  space, with  $\mathbf{J}$  defined as:

$$\mathbf{J} = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial g_1(\mathbf{x})}{\partial x_N} \\ \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial g_2(\mathbf{x})}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_N(\mathbf{x})}{\partial x_1} & \frac{\partial g_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial g_N(\mathbf{x})}{\partial x_N} \end{bmatrix}$$

When the mapping is not one-to-one, the derived PDF can be found as a summation of terms for each root of the equation  $\mathbf{y} = \mathbf{g}(\mathbf{x})$ , as for the univariate case.

### 3.13 The multivariate Gaussian

We will firstly present the multivariate Gaussian with independent elements and then use the derived PDF result to derive the general case.

Consider  $N$  independent random variables  $X_i$  with ‘standard’ (i.e. with mean zero and unity standard deviation) normal distributions:

$$f(x_i) = N(0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2}$$

Since the  $X_i$ ’s are independent we can immediately write the joint (vector) PDF for all  $N$  RVs as:

$$\begin{aligned} f(x_1, x_2, \dots, x_N) &= f(x_1)f(x_2) \dots f(x_N) \\ &= \prod_{i=1}^N \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} \\ &= \frac{1}{\sqrt{2\pi}^N} \exp\left(-\frac{1}{2} \sum_{i=1}^N x_i^2\right) \\ &= \frac{1}{\sqrt{2\pi}^N} \exp\left(-\frac{1}{2} \|\mathbf{x}\|_2^2\right) \end{aligned}$$

Now apply the general linear (invertible) matrix transformation to the vector random variable  $\mathbf{X}$ :

$$\mathbf{Y} = \mathbf{g}(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{b}, \quad \text{where } \mathbf{A} \text{ is an invertible } N \times N \text{ matrix}$$

We wish to determine the derived PDF  $f_{\mathbf{Y}}(\mathbf{y})$ . Firstly find the Jacobian,  $|\mathbf{J}| = \left| \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}} \right|$ :

$$\begin{aligned} y_i &= \sum_{j=1}^N [\mathbf{A}]_{ij} x_j + b_i \\ \therefore [\mathbf{J}]_{ij} &= \frac{\partial y_i}{\partial x_j} = [\mathbf{A}]_{ij} \\ \therefore \mathbf{J} &= \mathbf{A} \\ \therefore |\det(\mathbf{J})| &= |\det(\mathbf{A})| \end{aligned}$$

The matrix transformation is invertible, so we can write:

$$\mathbf{x} = \mathbf{g}^{-1}(\mathbf{y}) = \mathbf{A}^{-1}(\mathbf{y} - \mathbf{b})$$

Hence:

$$\begin{aligned} f_{\mathbf{Y}}(\mathbf{y}) &= \frac{1}{|\det(\mathbf{J})|} f_{\mathbf{X}}(\mathbf{x})|_{\mathbf{x}=\mathbf{g}^{-1}(\mathbf{y})} \\ &= \frac{1}{|\det(\mathbf{A})|} \frac{1}{\sqrt{2\pi}^N} \exp\left(-\frac{1}{2} \|\mathbf{g}^{-1}(\mathbf{y})\|_2^2\right) \\ &= \frac{1}{|\det(\mathbf{A})|} \frac{1}{\sqrt{2\pi}^N} \exp\left(-\frac{1}{2} \|\mathbf{A}^{-1}(\mathbf{y} - \mathbf{b})\|_2^2\right) \\ &= \frac{1}{|\det(\mathbf{A})|} \frac{1}{\sqrt{2\pi}^N} \exp\left(-\frac{1}{2} (\mathbf{y} - \mathbf{b})^T (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{y} - \mathbf{b})\right) \end{aligned}$$

Now define  $\mathbf{C}_{\mathbf{Y}} = \mathbf{A}^T \mathbf{A}$ , and hence  $|\det(\mathbf{A})| = |\det(\mathbf{C}_{\mathbf{Y}})|^{\frac{1}{2}}$ . Substituting  $\mathbf{C}_{\mathbf{Y}} = \mathbf{A}^T \mathbf{A}$  and  $\boldsymbol{\mu} = \mathbf{b}$  we obtain finally:

$$\boxed{f_{\mathbf{Y}}(\mathbf{y}) = \frac{1}{\sqrt{2\pi}^N |\det(\mathbf{C}_{\mathbf{Y}})|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^T \mathbf{C}_{\mathbf{Y}}^{-1} (\mathbf{y} - \boldsymbol{\mu})\right)}$$

*Multivariate Gaussian PDF*

where  $\boldsymbol{\mu} = E[\mathbf{Y}]$  is the mean vector and  $\mathbf{C}_{\mathbf{Y}} = E[(\mathbf{Y} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu})^T]$  is the *Covariance Matrix* (since  $[\mathbf{C}_{\mathbf{Y}}]_{ij} = c_{Y_i Y_j}$  the covariance between elements  $Y_i$  and  $Y_j$ ). Note that  $f_{\mathbf{X}}(\mathbf{x})$ , the PDF of the original independent data, is a special case of the multivariate Gaussian with  $\mathbf{C}_{\mathbf{Y}} = \mathbf{I}$ , the identity matrix. The multivariate Gaussian will be used extensively throughout the remainder of this book. See figures 3.12 and 3.12 for contour and 3-D plots of an example of the 2-dimensional Gaussian density.

### 3.14 Random signals

In this section we extend the ideas of previous sections to model the properties of *Random Signals*, i.e. waveforms which are of infinite duration and which can take on random values at all points in time.

Consider an underlying Probability Space  $(\Omega, \mathcal{F}, P)$ , as before. However, instead of mapping from the Sample Space  $\Omega$  to a finite point or set of points such as a random variable or random vector, each member  $\omega$  of the Sample Space now maps onto a real-valued *waveform*  $X(t, \omega)$ , with values possibly defined for all times  $t$ .

We can imagine a generalisation of our previous ideas about random experiments so that the outcome of an experiment can be a ‘Random Object’, an example of which is a waveform chosen at random (according to a Probability Measure  $P$ ) from a set of possible waveforms, which we term an *Ensemble*.

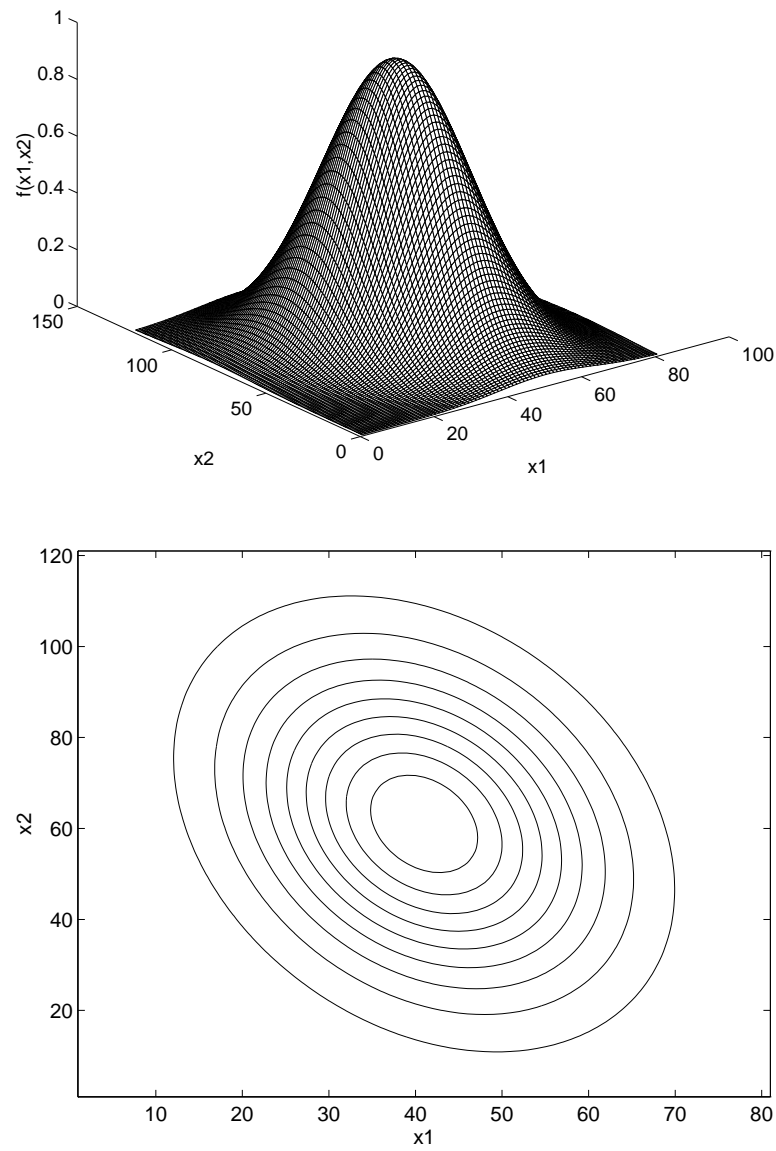


FIGURE 3.1. Example of the bivariate Gaussian PDF  $f_{X_1, X_2}(x_1, x_2)$ . Top - 3-D mesh plot, bottom - contour plot



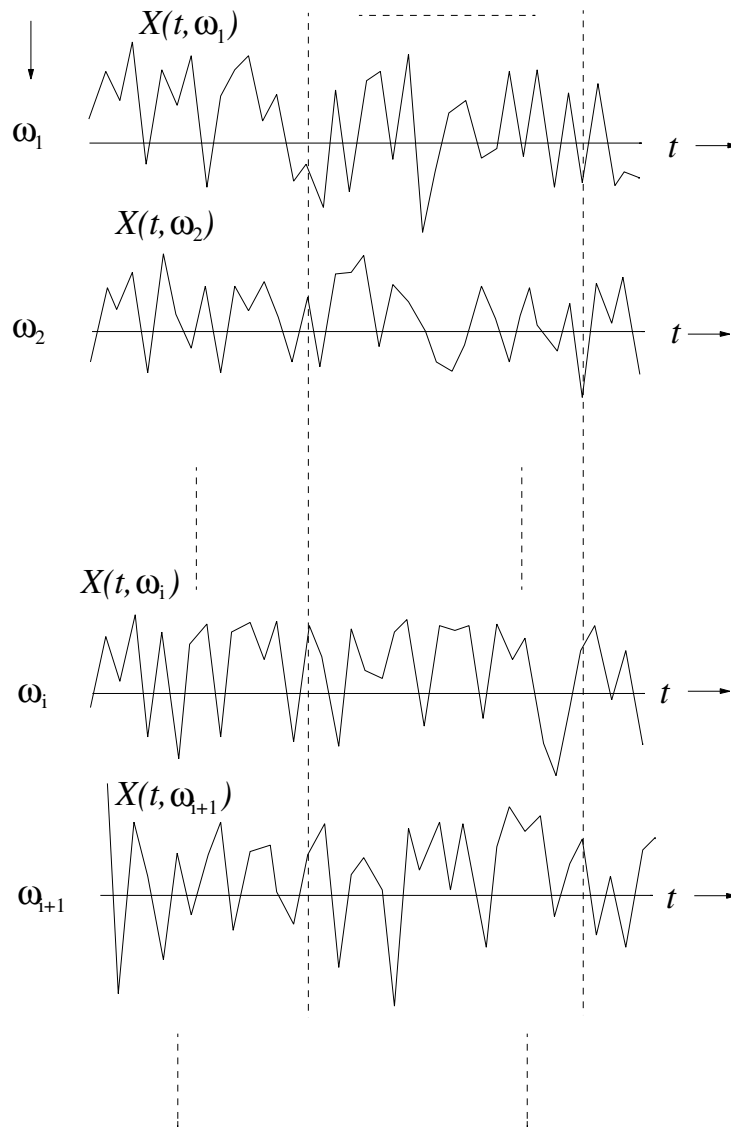


FIGURE 3.2. Ensemble representation of a random process

### 3.15 Definition: random process

Given a probability space  $(\Omega, \mathcal{F}, P)$ , a Random Process  $\{X(t, \omega)\}$  is a function which maps each element  $\omega$  of the sample space  $\Omega$  onto a real-valued function of time, see figure 3.14 The mapping is defined at times  $t$  belonging to some set  $\mathcal{T}$  (a more rigorous notation would thus be  $\{X(t, \omega); t \in \mathcal{T}\}$ ). If  $\mathcal{T}$  is a continuous set, e.g.  $\mathbb{R}$  or  $[0, \infty)$ , then the process is termed a *Continuous Time* Random Process. If  $\mathcal{T}$  is a discrete set of time values, e.g.  $\mathbb{Z}$ , the integers, the process is termed *Discrete Time* or a *Time Series*. We will be concerned mainly with discrete time processes in this book. As for random variables we will usually drop the explicit dependence on  $\omega$  for notational convenience, referring simply to ‘Random Process  $\{X(t)\}$ ’. If we consider the process  $\{X(t)\}$  at one particular time  $t = t_1$  then we have a Random Variable  $X(t_1)$ . If we consider the process  $\{X(t)\}$  at  $N$  time instants  $\{t_1, t_2, \dots, t_N\}$  then we have a Random Vector:

$$\mathbf{X} = \begin{bmatrix} X(t_1) \\ X(t_2) \\ \vdots \\ X(t_N) \end{bmatrix}$$

We can study the properties of a Random Process by considering the behaviour of random variables and random vectors extracted from the process at times  $t_i$ . We already have the means to achieve this through the theory of random variables and random vectors.

We will limit our discussion here to discrete time random processes or time series defined at time instants  $\mathcal{T} = \{t_{-\infty}, \dots, t_{-1}, t_0, t_1, \dots, t_{\infty}\}$ , where usually (although this is not a requirement) the process is *uniformly sampled*, i.e.  $t_n = nT$  where  $T$  is the sampling interval. We will use the notation  $X_n = X(t_n)$  for random variables extracted at the  $i$ th time point and the whole discrete time process will be referred to as  $\{X_n\}$ .

#### 3.15.1 Mean values and correlation functions

The mean value of a discrete time random process is defined as:

$$\boxed{\mu_n = E[X_n]}$$

*Mean value of random process*

and the discrete *autocorrelation* function is:

$$r_{XX}(n, m) = E[X_n X_m]$$

*Autocorrelation function of random process*

The cross-correlation function between two processes  $\{X_n\}$  and  $\{Y_n\}$  is:

$$r_{XY}(n, m) = E[X_n Y_m]$$

*Cross-correlation function*

### 3.15.2 Stationarity

A stationary process has the same statistical characteristics irrespective of time shifts along the axis. To put it another way, an observer looking at the process from time  $t_i$  would not be able to tell the difference in the statistical characteristics of the process if he moved to a different time  $t_j$ . This idea is formalised by considering the  $N$ th order density for the process:

$$f_{X_{n_1}, X_{n_2}, \dots, X_{n_N}}(x_{n_1}, x_{n_2}, \dots, x_{n_N})$$

*$N$ th order density for a random process*

which is the joint probability density function for  $N$  arbitrarily chosen time indices  $\{n_1, n_2, \dots, n_N\}$ . Since the probability distribution of a random vector contains all the statistical information about that random vector, we would expect the probability distribution to be unchanged if we shifted the time axis any amount to the left or the right. This is the idea behind *strict-sense stationarity* for a discrete random process.

A random process is strict-sense stationary if, *for any finite  $c$ ,  $N$  and  $\{n_1, n_2, \dots, n_N\}$ :*

$$\boxed{\begin{aligned} f_{X_{n_1}, X_{n_2}, \dots, X_{n_N}}(x_{n_1}, x_{n_2}, \dots, x_{n_N}) \\ = f_{X_{n_1+c}, X_{n_2+c}, \dots, X_{n_N+c}}(x_{n_1}, x_{n_2}, \dots, x_{n_N}) \end{aligned}}$$

*Strict-sense stationarity for a random process*

A less stringent condition which is nevertheless very useful for practical analysis is *wide-sense stationarity* which only requires first and second order moments to be invariant to time shifts.

A random process is wide-sense stationary if:

1.  $\mu_n = E[X_n] = \mu$ , (mean is constant)
2.  $r_{XX}(n, m) = r_{XX}(n - m)$ , (autocorrelation function depends only upon the difference between  $n$  and  $m$ ).

*Wide-sense stationarity for a random process*

Note that strict-sense stationarity implies wide-sense stationarity, but not *vice versa*.

### 3.15.3 Power spectra

For a wide-sense stationary random process  $\{X_n\}$ , the power spectrum is defined as the discrete-time Fourier transform (DTFT) of the discrete autocorrelation function:

$$\boxed{\mathcal{S}_X(\omega) = \sum_{m=-\infty}^{\infty} r_{XX}(m) e^{-jm\omega T}} \quad (3.6)$$

*Power spectrum for a random process*

and the autocorrelation function can thus be found from the power spectrum by inverting the transform:

$$r_{XX}(m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{S}_X(\omega) e^{jm\omega T} d\omega T \quad (3.7)$$

*Autocorrelation function from power spectrum*

The power spectrum is a real, even and periodic function of frequency. The power spectrum can be interpreted as a density spectrum in the sense that the expected power at the output of an ideal band-pass filter with lower and upper cut-off frequencies of  $\omega_l$  and  $\omega_u$  is given by

$$\frac{1}{\pi} \int_{\omega_l}^{\omega_u} \mathcal{S}_X(\omega) d\omega$$

#### 3.15.4 Linear systems and random processes

When a wide-sense stationary discrete random process  $\{X_n\}$  is passed through a linear time invariant (LTI) system with pulse response  $\{h_n\}$ , the output process  $\{Y_n\}$  is also wide-sense stationary and we can express the output correlation functions and power spectra in terms of the input statistics and the LTI system:

$$r_{YY}(l) = \sum_{k=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} h_k h_i r_{XX}(l + i - k) \quad (3.8)$$

*Autocorrelation function at the output of a LTI system*

$$\mathcal{S}_Y(\omega) = |H(e^{j\omega T})|^2 \mathcal{S}_X(\omega) \quad (3.9)$$

*Power spectrum at the output of a LTI system*

## 3.16 Conclusion

We have provided a brief survey of results from probability theory and random signal theory on which the remaining chapters are based. The next chapter describes the theory of detection and estimation.



# 4

## Parameter Estimation, Model Selection and Classification

In this chapter we review the fundamental techniques and principles behind estimation of unobserved variables from observed data ('parameter estimation') and selection of a classification or form of model to represent the observed data ('classification' and 'model selection'). The term parameter estimation is used in a very general sense here. For example, we will treat both the interpolation of missing data samples in a digitised waveform and the more obvious task of coefficient estimation for some parametric data model as examples of parameter estimation. Classification is traditionally concerned with choosing from a set of possible classes the class which best represents the observed data. This concept will be extended to include directly analogous tasks such as identifying which samples in a data sequence are corrupted by some intermittent noise process. Model selection is another type of classification in which the classes are a number of different model-based formulations for data and it is required to choose the best model for the observed data.

The last sections of the chapter introduce two related topics which will be important in later work. The first is Sequential Bayesian Classification in which classification estimates are updated sequentially with the input of new data samples. The second is autoregressive (AR) modelling considered within a Bayesian framework, which will be a fundamental part of subsequent chapters. We then discuss state-space models and the Kalman filter, an efficient means for implementing many of the sequential schemes, and finally introduce some sophisticated methods for exploration of posterior distributions: the expectation-maximisation and Markov chain Monte Carlo methods.

In all of our work parameters and models are treated as random variables or random vectors. We are thus prepared to specify prior information concerning the processes in a probabilistic form, which leads to a Bayesian approach to the solution of these problems. The chapter is thus primarily concerned with developing the principles of Bayesian Decision Theory and Estimation as applied to signal processing. We do not intend to provide a complete review of non-Bayesian techniques, although traditional methods are referenced and described and the relationship of Bayesian methods to Maximum Likelihood (ML) and Least Squares (LS) is discussed.

## 4.1 Parameter estimation

In parameter estimation we suppose that a random process  $\{X_n\}$  depends in some well-defined stochastic manner upon an unobserved parameter vector  $\boldsymbol{\theta}$ . If we observe  $N$  data points from the random process, we can form a vector  $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T$ . The parameter estimation problem is to deduce the value of  $\boldsymbol{\theta}$  from the observations  $\mathbf{x}$ . In general it will not be possible to deduce the parameters exactly from a finite number of data points since the process is random, but various schemes will be described which achieve different levels of performance depending upon the amount of data available and the amount of prior information available regarding  $\boldsymbol{\theta}$ .

### 4.1.1 The general linear model

We now define a general parametric model which will be referred to frequently in this and subsequent chapters. In this model it is assumed that the data  $\mathbf{x}$  are generated as a function of the parameters  $\boldsymbol{\theta}$  with an additive random modelling error term  $e_n$ :

$$x_n = g_n(\boldsymbol{\theta}) + e_n$$

where  $g_n(\cdot)$  is a deterministic and possibly non-linear function. For most of the models we will consider,  $g_n(\cdot)$  is a linear function of the parameters so we may write

$$x_n = \mathbf{g}_n^T \boldsymbol{\theta} + e_n$$

where  $\mathbf{g}_n$  is a  $P$ -dimensional column vector, and the expression may be written for the whole vector  $\mathbf{x}$  as

$$\boxed{\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{e}} \tag{4.1}$$

*General linear model*



where

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1^T \\ \mathbf{g}_2^T \\ \vdots \\ \mathbf{g}_N^T \end{bmatrix}$$

The columns of  $\mathbf{G}$  form a fixed basis vector representation of the data, for example sinusoids of different frequencies in a signal which is known to be made up of pure frequency components in noise. A variant of this model will be seen later in the chapter, the autoregressive (AR) model, in which previous data points are used to predict the current data point  $x_n$ . The error sequence  $\mathbf{e}$  will usually (but not necessarily) be assumed drawn from an independent, identically distributed (i.i.d.) noise distribution, that is,

$$p(\mathbf{e}) = p_e(e_1) \cdot p_e(e_2) \cdot \dots \cdot p_e(e_N)$$

where  $p_e(\cdot)$  denotes some noise distribution which is identical for all  $n$ .<sup>1</sup>  $\{e_n\}$  can be viewed as a modelling error, innovation or observation noise, depending upon the type of model. We will encounter the linear form (4.1) and its variants throughout the book, so it is used as the primary example in the subsequent sections of this chapter.

#### 4.1.2 Maximum likelihood (ML) estimation

The first method of estimation considered is the maximum likelihood (ML) estimator which treats the parameters as unknown constants about which we incorporate no prior information. The observed data  $\mathbf{x}$  is, however, considered random and we can often then obtain the PDF for  $\mathbf{x}$  when the value of  $\boldsymbol{\theta}$  is known. This PDF is termed the *likelihood*  $L(\mathbf{x}; \boldsymbol{\theta})$ , which is defined as

$$L(\mathbf{x}; \boldsymbol{\theta}) = p(\mathbf{x} | \boldsymbol{\theta}) \quad (4.2)$$

The likelihood is of course implicitly conditioned upon all of our modelling assumptions  $\mathcal{M}$ , which could be expressed as  $p(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M})$ . We do not adopt this convention, however, for reasons of notational simplicity.

The ML estimate for  $\boldsymbol{\theta}$  is then that value of  $\boldsymbol{\theta}$  which maximises the likelihood for given observations  $\mathbf{x}$ :

---

<sup>1</sup>Whenever the context makes it unambiguous we will adopt from now on a notation  $p(\cdot)$  to denote both probability density functions (PDFs) and probability mass functions (PMFs) for random variables and vectors.

$$\boxed{\boldsymbol{\theta}^{\text{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\{p(\mathbf{x} | \boldsymbol{\theta})\}} \quad (4.3)$$

*Maximum likelihood (ML) estimator*

The rationale behind this is that the ML solution corresponds to the parameter vector which would have generated the observed data  $\mathbf{x}$  with highest probability. The maximisation task required for ML estimation can be achieved using standard differential calculus for well-behaved and differentiable likelihood functions, and it is often convenient analytically to maximise the log-likelihood function  $l(\mathbf{x}; \boldsymbol{\theta}) = \log(L(\mathbf{x}; \boldsymbol{\theta}))$  rather than  $L(\mathbf{x}; \boldsymbol{\theta})$  itself. Since log is a monotonically increasing function the two solutions are identical.

In data analysis and signal processing applications the likelihood function is arrived at through knowledge of the stochastic model for the data. For example, in the case of the general linear model (4.1) the likelihood can be obtained easily if we know the form of  $p(\mathbf{e})$ , the joint PDF for the components of the error vector. The likelihood  $p_{\mathbf{x}|\boldsymbol{\theta}}$  is then found from a transformation of variables  $\mathbf{e} \rightarrow \mathbf{x}$  where  $\mathbf{e} = \mathbf{x} - \mathbf{G}\boldsymbol{\theta}$ . The Jacobian (see section 3.12) for this transformation is unity, so the likelihood is:

$$L(\mathbf{x}; \boldsymbol{\theta}) = p(\mathbf{x} | \boldsymbol{\theta}) = p_{\mathbf{e}}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) \quad (4.4)$$

The elements of the error vector  $\mathbf{e}$  are often assumed to be i.i.d. and Gaussian. If the variance of the Gaussian is  $\sigma_e^2$  then we have:

$$p_{\mathbf{e}}(\mathbf{e}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\right)$$

Such an assumption is a reflection of the fact that many naturally occurring processes are Gaussian and that many of the subsequent results are easily obtained in closed form. The likelihood is then

$$L(\mathbf{x}; \boldsymbol{\theta}) = p_{\mathbf{e}}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})\right) \quad (4.5)$$

which leads to the following log-likelihood expression:

$$\begin{aligned} l(\mathbf{x}; \boldsymbol{\theta}) &= -(N/2) \log(2\pi\sigma_e^2) - \frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) \\ &= -(N/2) \log(2\pi\sigma_e^2) - \frac{1}{2\sigma_e^2} \sum_{n=1}^N (x_n - \mathbf{g}_n^T \boldsymbol{\theta})^2 \end{aligned}$$

Maximisation of this function w.r.t.  $\boldsymbol{\theta}$  is equivalent to minimising the sum-squared of the error sequence  $E = \sum_{n=1}^N (x_n - \mathbf{g}_n^T \boldsymbol{\theta})^2$ . This is exactly the criterion which is applied in the familiar *least squares* (LS) estimation method. The ML estimator is obtained by taking derivatives w.r.t.  $\boldsymbol{\theta}$  and equating to zero:

$$\boxed{\boldsymbol{\theta}^{\text{ML}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{x}} \quad (4.6)$$

*Maximum likelihood for the general linear model*

which is, as expected, the familiar linear least squares estimate for model parameters calculated from finite length data observations. Thus we see that the ML estimator under the i.i.d. Gaussian error assumption is exactly equivalent to the least squares (LS) solution.

### 4.1.3 Bayesian estimation

Recall that ML methods treat parameters as unknown constants. If we are prepared to treat parameters as random variables it is possible to assign prior PDFs to the parameters. These PDFs should ideally express some prior knowledge about the relative probability of different parameter values *before the data are observed*. Of course if nothing is known *a priori* about the parameters then the prior distributions should in some sense express no initial preference for one set of parameters over any other. Note that in many cases a prior density is chosen to express some highly qualitative prior knowledge about the parameters. In such cases the prior chosen will be more a reflection of a *degree of belief* [97] concerning parameter values than any true modelling of an underlying random process which might have generated those parameters. This willingness to assign priors which reflect subjective information is a powerful feature and also one of the most fundamental differences between the Bayesian and ‘classical’ inferential procedures. For various expositions of the Bayesian methodology and philosophy see, for example [97, 23, 15]. We take here a pragmatic viewpoint: if we have prior information about a problem then we will generally get better results through careful incorporation of that information. The precise form of probability distributions assigned *a priori* to the parameters requires careful consideration since misleading results can be obtained from erroneous priors, but in principle at least we can apply the Bayesian approach to any problem where statistical uncertainty is present.

Bayes rule is now stated as applied to estimation of random parameters  $\boldsymbol{\theta}$  from a random vector  $\mathbf{x}$  of observations, known as the posterior or *a posteriori* probability for the parameter:

$$\boxed{p(\boldsymbol{\theta} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathbf{x})}} \quad (4.7)$$

*Posterior probability*

Note that all of the distributions in this expression are implicitly conditioned upon all prior modelling assumptions, as was the likelihood function earlier. The distribution  $p(\mathbf{x} \mid \boldsymbol{\theta})$  is the likelihood as used for ML estimation, while  $p(\boldsymbol{\theta})$  is the prior or *a priori* distribution for the parameters. This term is one of the critical differences between Bayesian and ‘classical’ techniques. It expresses in an objective fashion the probability of various model parameters values *before* the data  $\mathbf{x}$  has been observed. As we have already observed, the prior density may be an expression of highly subjective information about parameter values. This transformation from the subjective domain to an objective form for the prior can clearly be of great significance and should be considered carefully.

The term  $p(\boldsymbol{\theta} \mid \mathbf{x})$ , the posterior or *a posteriori* distribution, expresses the probability of  $\boldsymbol{\theta}$  given the observed data  $\mathbf{x}$ . This is now a true measure of how ‘probable’ a particular value of  $\boldsymbol{\theta}$  is, given the observations  $\mathbf{x}$ .  $p(\boldsymbol{\theta} \mid \mathbf{x})$  is in a more intuitive form for parameter estimation than the likelihood, which expresses how probable the *observations* are given the *parameters*. The generation of the posterior distribution from the prior distribution when data  $\mathbf{x}$  is observed can be thought of as a refinement to any previous (‘prior’) knowledge about the parameters. Before  $\mathbf{x}$  is observed  $p(\boldsymbol{\theta})$  expresses any information previously obtained concerning  $\boldsymbol{\theta}$ . Any new information concerning the parameters contained in  $\mathbf{x}$  is then incorporated to give the posterior distribution. Clearly if we start off with little or no information about  $\boldsymbol{\theta}$  then the posterior distribution is likely to obtain information obtained almost solely from  $\mathbf{x}$ . Conversely, if  $p(\boldsymbol{\theta})$  expresses a significant amount of information about  $\boldsymbol{\theta}$  then  $\mathbf{x}$  will contribute less new information to the posterior distribution.

The denominator  $p(\mathbf{x})$ , sometimes referred to as the ‘evidence’ because of its interpretation in model selection problems (see later), is constant for any given observation  $\mathbf{x}$  and so may be ignored if we are only interested in the relative posterior probabilities of different parameters. In fact, Bayes rule is often stated in the form:

$$\boxed{p(\boldsymbol{\theta} \mid \mathbf{x}) \propto p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})} \quad (4.8)$$

*Posterior probability (proportionality)*

$p(\mathbf{x})$  may be calculated, however, by integration:

$$p(\mathbf{x}) = \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (4.9)$$

and this effectively serves as the normalising constant for the posterior density (in this and subsequent results the integration would be replaced by a summation in the case of a discrete random vector  $\boldsymbol{\theta}$ ).

#### 4.1.3.1 Posterior inference and Bayesian cost functions

The posterior distribution gives the probability for any chosen  $\boldsymbol{\theta}$  given observed data  $\mathbf{x}$ , and as such optimally combines our prior information about  $\boldsymbol{\theta}$  and any additional information gained about  $\boldsymbol{\theta}$  from observing  $\mathbf{x}$ . We may in principle manipulate the posterior density to infer any required statistic of  $\boldsymbol{\theta}$  conditional upon  $\mathbf{x}$ . This is an advantage over ML and least squares methods which strictly give us only a single estimate of  $\boldsymbol{\theta}$ , known as a ‘point estimate’. However, by producing a posterior PDF with values defined for all  $\boldsymbol{\theta}$  the Bayesian approach gives a fully interpretable probability distribution. In principle this is as much as one could ever need to know. In signal processing problems, however, we usually require a single point estimate for  $\boldsymbol{\theta}$ , and a suitable way to choose this is via a ‘cost function’  $C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$  which expresses objectively a measure of the cost associated with a particular parameter estimate  $\hat{\boldsymbol{\theta}}$  when the true parameter is  $\boldsymbol{\theta}$  (see e.g. [51, 15]). The form of cost function will depend on the requirements of a particular problem. A cost of 0 indicates that the estimate is perfect for our requirements (this does not necessarily imply that  $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$ ) while positive values indicate poorer estimates. The *risk* associated with a particular estimator is then defined as the expected posterior cost associated with that estimate:

$$R(\hat{\boldsymbol{\theta}}) = E[C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})] = \int_{\mathbf{x}} \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{x}) p(\mathbf{x}) d\boldsymbol{\theta} d\mathbf{x}. \quad (4.10)$$

We require the estimation scheme which chooses  $\hat{\boldsymbol{\theta}}$  to minimise the risk. The minimum risk is known as the ‘Bayes risk’. For non-negative cost functions it is sufficient to minimise only the inner integral

$$I(\hat{\boldsymbol{\theta}}) = \int_{\boldsymbol{\theta}} C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{x}) d\boldsymbol{\theta} \quad (4.11)$$

for all  $\hat{\boldsymbol{\theta}}$ . Typical cost functions are the quadratic cost function  $|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|^2$  and the uniform cost function, defined for arbitrarily small  $\varepsilon$  as

$$C(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \begin{cases} 1, & |\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}| > \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (4.12)$$

The quadratic cost function leads to the minimum mean-squared error (MMSE) estimator and as such is reasonable for many examples of parameter estimation, where we require an estimate representative of the whole posterior density. The MMSE estimate can be shown to equal the *mean* of the posterior distribution:

$$\hat{\boldsymbol{\theta}} = \int_{\boldsymbol{\theta}} \boldsymbol{\theta} p(\boldsymbol{\theta} | \mathbf{x}) d\boldsymbol{\theta}. \quad (4.13)$$

Where the posterior distribution is symmetrical about its mean the posterior mean can in fact be shown to be the minimum risk solution for any cost function which is a convex function of  $|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|$  [186].

The uniform cost function is useful for the ‘all or nothing’ scenario where we wish to attain the correct parameter estimate at all costs and any other estimate is of no use. Therrien [174] cites the example of a pilot landing a plane on an aircraft carrier. If he does not estimate within some small finite error he misses the ship, in which case the landing is a disaster. The uniform cost function for  $\varepsilon \rightarrow 0$  leads to the maximum *a posteriori* (MAP) estimate, the value of  $\hat{\boldsymbol{\theta}}$  which maximises the posterior distribution:

$$\boxed{\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \{p(\boldsymbol{\theta} | \mathbf{x})\}} \quad (4.14)$$

*Maximum a posteriori (MAP) estimator*

Note that for Gaussian posterior distributions the MMSE and MAP solutions coincide, as indeed they do for any distribution symmetric about its mean with its maximum at the mean. The MAP estimate is thus appropriate for many common estimation problems, including those encountered in this book.

We now work through the MAP estimation scheme for the general linear model (4.1). Suppose that the prior on parameter vector  $\boldsymbol{\theta}$  is the multivariate Gaussian (A.2):

$$p(\boldsymbol{\theta}) = \frac{1}{(2\pi)^{P/2} |\mathbf{C}_{\boldsymbol{\theta}}|^{1/2}} \exp \left( -\frac{1}{2} (\boldsymbol{\theta} - \mathbf{m}_{\boldsymbol{\theta}})^T \mathbf{C}_{\boldsymbol{\theta}}^{-1} (\boldsymbol{\theta} - \mathbf{m}_{\boldsymbol{\theta}}) \right) \quad (4.15)$$

where  $\mathbf{m}_\theta$  is the parameter mean vector,  $\mathbf{C}_\theta$  is the parameter covariance matrix and  $P$  is the number of parameters in  $\theta$ . If the likelihood  $p(\mathbf{x} | \theta)$  takes the same form as before (4.5), the posterior distribution is as follows:

$$p(\theta | \mathbf{x}) \propto \frac{1}{(2\pi)^{P/2} |\mathbf{C}_\theta|^{1/2}} \frac{1}{(2\pi\sigma_e^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x} - \mathbf{G}\theta)^T(\mathbf{x} - \mathbf{G}\theta) - \frac{1}{2}(\theta - \mathbf{m}_\theta)^T \mathbf{C}_\theta^{-1}(\theta - \mathbf{m}_\theta)\right)$$

and the MAP estimate  $\theta^{\text{MAP}}$  is obtained by differentiation of the log-posterior as:

$$\boxed{\theta^{\text{MAP}} = (\mathbf{G}^T \mathbf{G} + \sigma_e^2 \mathbf{C}_\theta^{-1})^{-1} (\mathbf{G}^T \mathbf{x} + \sigma_e^2 \mathbf{C}_\theta^{-1} \mathbf{m}_\theta)} \quad (4.16)$$

*MAP estimator - general linear model*

In this expression we can clearly see the ‘regularising’ effect of the prior density on the ML estimate of (4.6). As the prior becomes more ‘diffuse’, i.e. the diagonal elements of  $\mathbf{C}_\theta$  increase both in magnitude and relative to the off-diagonal elements, we impose ‘less’ prior information on the estimate. In the limit the prior tends to a uniform (‘flat’) prior with all  $\theta$  equally probable. In this limit  $\mathbf{C}_\theta^{-1} = 0$  and the estimate is identical to the ML estimate (4.6). This important relationship demonstrates that the ML estimate may be interpreted as the MAP estimate with a uniform prior assigned to  $\theta$ . The MAP estimate will also tend towards the ML estimate when the likelihood is strongly ‘peaked’ around its maximum compared with the prior. Once again the prior will have little influence on the estimate. It is in fact well known [97] that as the sample size  $N$  tends to infinity the Bayes solution tends to the ML solution. This of course says nothing about small sample parameter estimates where the effect of the prior may be very significant.

#### 4.1.3.2 Marginalisation for elimination of unwanted parameters

In many cases we can formulate a likelihood function for a particular problem which depends on more unknown parameters than are actually wanted for estimation. These will often be ‘scale’ parameters such as unknown noise or excitation variances but may also be unobserved (‘missing’) data values or unwanted system parameters. A full ML procedure requires that the

likelihood be maximised w.r.t. all of these parameters and the unwanted values are then simply discarded to give the required estimate.<sup>2</sup> However, this may not in general be an appropriate procedure for obtaining only the required parameters - a cost function which depends only upon a certain subset of parameters leads to an estimator which only depends upon the *marginal* probability for those parameters. The Bayesian approach allows for the interpretation of these unwanted or ‘nuisance’ parameters as random variables, for which as usual we can specify prior densities. The marginalisation identity can be used to eliminate these parameters from the posterior distribution, and from this we are able to obtain a posterior distribution in terms of only the desired parameters. Consider an unwanted parameter  $\phi$  which is present in the modelling assumptions. The unwanted parameter is now eliminated from the posterior expression by marginalisation:

$$\begin{aligned} p(\boldsymbol{\theta}|\mathbf{x}) &= \int_{\phi} p(\boldsymbol{\theta}, \phi|\mathbf{x}) d\phi \\ &\propto \int_{\phi} p(\mathbf{x}|\boldsymbol{\theta}, \phi)p(\boldsymbol{\theta}, \phi) d\phi \end{aligned} \quad (4.17)$$

#### 4.1.3.3 Choice of priors

One of the major criticisms levelled at Bayesian techniques is that the choice of priors can be highly subjective and, as mentioned above, the inclusion of inappropriate priors could give misleading results. There is clearly no problem if the prior statistics are genuinely known. Difficulties arise, however, when an attempt is made to choose prior densities to express some very subjective piece of prior knowledge or when nothing at all appears to be known beforehand about the parameters.

In many engineering applications we may have genuine prior belief in terms of the observed long term behaviour of a particular parameter before the current data were observed. In these cases there is clearly very little problem with the Bayesian approach. In other cases subjective information may be available, such as an approximate range of feasible values, or a rough idea of the parameter’s likely value. In these cases a pragmatic approach can be adopted: choose a prior with convenient analytical properties which expresses the type of information available to us. We have already seen an example of such a prior in the Gaussian prior assigned to the linear parameter vector (4.15). If an approximate value for  $\boldsymbol{\theta}$  were known, this could be chosen as the mean vector  $\mathbf{m}_{\boldsymbol{\theta}}$  of the Gaussian. If we can quantify our uncertainty about this mean value then this can be incorporated into the prior covariance matrix. To take another example, the variance of

---

<sup>2</sup>Although in time series likelihood-based analysis it is common to treat missing and other unobserved data in a Bayesian fashion, see [91] and references therein.



the error  $\sigma_e^2$  might be unknown. However, physical considerations for the system might lead us to expect that a certain range of values is more likely than others. A suitable prior for this case might be the *inverted Gamma* distribution (A.11), which is defined for positive random variables and can be assigned a mean and variance. In either of these cases the Bayesian analysis of the linear model remains analytically tractable and such priors are termed *conjugate* priors (see [15, appendix A.2] for a comprehensive list of likelihood functions and their associated conjugate priors).

In cases where no prior information can be assumed we will wish to assign a prior which does not bias the outcome of the inference procedure. Such priors are termed *non-informative* and detailed discussions can be found in Box and Tiao [23] and Bernardo and Smith [15]. We do not detail the construction of non-informative priors here, but give two important examples which are much used. Firstly, a non-informative prior which is commonly used for the linear ‘location’ parameters  $\boldsymbol{\theta}$  in the general linear model (4.1) is the uniform prior. Secondly, the standard non-informative prior for a ‘scale’ parameter such as  $\sigma_e^2$  in the linear model, is the Jeffreys’ prior [97] which has the form  $p(\sigma^2) = \frac{1}{\sigma^2}$ . These priors are designed to give invariance of the inferential procedure to re-scaling of the data. A troublesome point, however, is that both are unnormalised, which means they are not proper densities. Furthermore, there are some models for which it is not possible to construct a non-informative prior.

#### 4.1.4 Bayesian Decision theory

As for Bayesian parameter estimation we consider the unobserved variable (in this case a classification state  $s_i$ ) as being generated by some random process whose prior probabilities are known. These prior probabilities are assigned to each of the possible classification states using a probability mass function (PMF)  $p(s_i)$ , which expresses the prior probability of occurrence of different states given all information available except the data  $\mathbf{x}$ . The required form of Bayes rule for this discrete estimation problem is then

$$p(s_i | \mathbf{x}) = \frac{p(\mathbf{x} | s_i) p(s_i)}{p(\mathbf{x})} \quad (4.18)$$

$p(\mathbf{x})$  is constant for any given  $\mathbf{x}$  and will serve to normalise the posterior probabilities over all  $i$  in the same way that the ‘evidence’ normalised the posterior parameter distribution (4.7). In the same way that Bayes rule gave a posterior distribution for parameters  $\boldsymbol{\theta}$ , this expression gives the posterior probability for a particular state given the observed data  $\mathbf{x}$ . It would seem reasonable to choose the state  $s_i$  corresponding to maximum posterior probability as our estimate for the true state (we will refer to this state estimate as the MAP estimate), and this can be shown to have the desirable property of minimum classification error rate  $P_E$  (see e.g. [51]),

that is, it has minimum probability of choosing the wrong state. Note that determination of the MAP state vector will usually involve an exhaustive search of  $p(s_i | \mathbf{x})$  for all feasible  $i$ , although sub-optimal schemes are developed for specific applications encountered later.

These ideas are formalised by consideration of a ‘loss function’  $\lambda(\alpha_i | s_j)$  which defines the penalty incurred by taking action  $\alpha_i$  when the true state is  $s_j$ . Action  $\alpha_i$  will usually refer to the action of choosing state  $s_i$  as the state estimate.

The expected risk associated with action  $\alpha_i$  (known as the conditional risk) is then expressed as

$$R(\alpha_i | \mathbf{x}) = \sum_{j=1}^{N_s} \lambda(\alpha_i | s_j) p(s_j | \mathbf{x}). \quad (4.19)$$

It can be shown that it is sufficient to minimise this conditional risk in order to achieve the optimal decision rule for a given problem and loss function.

Consider a loss function which is zero when  $i = j$  and unity otherwise. This ‘symmetric’ loss function can be viewed as the equivalent of the uniform cost function used for parameter estimation (4.12). The conditional risk is then given by:

$$R(\alpha_i | \mathbf{x}) = \sum_{j=1, (j \neq i)}^{N_s} p(s_j | \mathbf{x}) \quad (4.20)$$

$$= 1 - p(s_i | \mathbf{x}) \quad (4.21)$$

The second line here is simply the conditional probability that action  $\alpha_i$  is *incorrect*, and hence minimisation of the conditional risk is equivalent to minimisation of the probability of classification error,  $P_E$ . It is clear from this expression that selection of the MAP state is the optimal decision rule for the symmetric loss function.

Cases where minimum classification error rate may not be the best criterion are encountered and discussed later. However, alternative loss functions will not in general lead to estimators as simple as the MAP estimate.

#### 4.1.4.1 Calculation of the evidence, $p(\mathbf{x} | s_i)$

The term  $p(\mathbf{x} | s_i)$  is equivalent to the ‘evidence’ term  $p(\mathbf{x})$  which was encountered in the parameter estimation section, since  $p(\mathbf{x})$  was implicitly conditioned on a particular model structure or state in that scheme.

If one uses a uniform state prior  $p(s_i) = \frac{1}{N_s}$ , then, according to equation (4.18), it is only necessary to compare values of  $p(\mathbf{x} | s_i)$  for model selection since the remaining terms are constant for all models.  $p(\mathbf{x} | s_i)$  can then be viewed literally as the relative ‘evidence’ for a particular model, and two candidate models can be compared through their Bayes Factor:

$$BF_{ij} = \frac{p(\mathbf{x} | s_i)}{p(\mathbf{x} | s_j)}$$

Typically each model or state  $s_i$  will be expressed in a parametric form whose parameters  $\boldsymbol{\theta}_i$  are unknown. As for the parameter estimation case it will usually be possible to obtain the state conditional parameter likelihood  $p(\mathbf{x} | \boldsymbol{\theta}_i, s_i)$ . Given a state-dependent prior distribution for  $\boldsymbol{\theta}_i$  the evidence may be obtained by integration to eliminate  $\boldsymbol{\theta}_i$  from the joint probability  $p(\mathbf{x}, \boldsymbol{\theta}_i | s_i)$ . The evidence is then obtained using result (4.17) as

$$p(\mathbf{x} | s_i) = \int_{\boldsymbol{\theta}_i} p(\mathbf{x} | \boldsymbol{\theta}_i, s_i) p(\boldsymbol{\theta}_i | s_i) d\boldsymbol{\theta}_i \quad (4.22)$$

If the general linear model of (4.1) is extended to the multi-model scenario we obtain:

$$\mathbf{x} = \mathbf{G}_i \boldsymbol{\theta}_i + \mathbf{e}_i \quad (4.23)$$

where  $\mathbf{G}_i$  refers to the state-dependent basis matrix and  $\mathbf{e}_i$  is the corresponding error sequence. For this model the state dependent parameter likelihood  $p(\mathbf{x} | \boldsymbol{\theta}_i, s_i)$  is (see (4.5)):

$$p(\mathbf{x} | \boldsymbol{\theta}_i, s_i) = \frac{1}{(2\pi\sigma_{e_i}^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_{e_i}^2}(\mathbf{x} - \mathbf{G}_i \boldsymbol{\theta}_i)^T (\mathbf{x} - \mathbf{G}_i \boldsymbol{\theta}_i)\right) \quad (4.24)$$

and assuming the same Gaussian form for the state conditional parameter prior  $p(\boldsymbol{\theta}_i | s_i)$  (with  $P_i$  parameters) as we used for  $p(\boldsymbol{\theta})$  in (4.15) the evidence is then given by:

$$\begin{aligned} p(\mathbf{x} | s_i) = \int_{\boldsymbol{\theta}_i} & \frac{1}{(2\pi)^{P_i/2} |\mathbf{C}_{\boldsymbol{\theta}_i}|^{1/2}} \frac{1}{(2\pi\sigma_{e_i}^2)^{N/2}} \\ & \exp\left(-\frac{1}{2\sigma_{e_i}^2}(\mathbf{x} - \mathbf{G}_i \boldsymbol{\theta}_i)^T (\mathbf{x} - \mathbf{G}_i \boldsymbol{\theta}_i) \right. \\ & \left. - \frac{1}{2}(\boldsymbol{\theta}_i - \mathbf{m}_{\boldsymbol{\theta}_i})^T \mathbf{C}_{\boldsymbol{\theta}_i}^{-1} (\boldsymbol{\theta}_i - \mathbf{m}_{\boldsymbol{\theta}_i})\right) d\boldsymbol{\theta}_i \end{aligned} \quad (4.25)$$

This multivariate Gaussian integral can be performed after some rearrangement using result (A.5) to give:

$$\begin{aligned} p(\mathbf{x} | s_i) = & \frac{1}{(2\pi)^{P_i/2} |\mathbf{C}_{\boldsymbol{\theta}_i}|^{1/2} |\Phi|^{1/2} (2\pi\sigma_{e_i}^2)^{(N-P_i)/2}} \\ & \exp\left(-\frac{1}{2\sigma_{e_i}^2}(\mathbf{x}^T \mathbf{x} + \sigma_{e_i}^2 \mathbf{m}_{\boldsymbol{\theta}_i}^T \mathbf{C}_{\boldsymbol{\theta}_i}^{-1} \mathbf{m}_{\boldsymbol{\theta}_i} - \boldsymbol{\Theta}^T \boldsymbol{\theta}_i^{\text{MAP}})\right) \end{aligned} \quad (4.26)$$

with terms defined as

$$\boldsymbol{\theta}_i^{\text{MAP}} = \Phi^{-1} \boldsymbol{\Theta} \quad (4.27)$$

$$\Phi = \mathbf{G}_i^T \mathbf{G}_i + \sigma_{e_i}^2 \mathbf{C}_{\boldsymbol{\theta}_i}^{-1} \quad (4.28)$$

$$\boldsymbol{\Theta} = \mathbf{G}_i^T \mathbf{x} + \sigma_{e_i}^2 \mathbf{C}_{\boldsymbol{\theta}_i}^{-1} \mathbf{m}_{\boldsymbol{\theta}_i} \quad (4.29)$$

Notice that  $\boldsymbol{\theta}_i^{\text{MAP}}$  is simply the state dependent version of the MAP parameter estimate given by (4.16).

This expression for the state evidence is proportional to the joint probability  $p(\mathbf{x}, \boldsymbol{\theta}_i | s_i)$  with the MAP value  $\boldsymbol{\theta}_i^{\text{MAP}}$  substituted for  $\boldsymbol{\theta}_i$ . This is easily verified by substitution of  $\boldsymbol{\theta}_i^{\text{MAP}}$  into the integrand of (4.25) and comparing with (4.26). The following relationship results:

$$p(\mathbf{x} | s_i) = \frac{(2\pi\sigma_{e_i}^2)^{P_i/2}}{|\Phi|^{1/2}} p(\mathbf{x}, \boldsymbol{\theta}_i | s_i) |_{\boldsymbol{\theta}_i = \boldsymbol{\theta}_i^{\text{MAP}}} . \quad (4.30)$$

Thus within this all-Gaussian framework the only difference between substituting in MAP parameters for  $\boldsymbol{\theta}_i$  and performing the full marginalisation integral is a multiplicative term  $\frac{(2\pi\sigma_{e_i}^2)^{P_i/2}}{|\Phi|^{1/2}}$ .

In the case where the state dependent likelihood is expressed in terms of unknown scale parameters (e.g.  $\sigma_{e_i}^2$ ) it is sometimes possible to marginalise these parameters in addition to the system parameters  $\boldsymbol{\theta}_i$ , depending upon the prior structure chosen. The same conjugate prior as proposed in section 4.1.3.3, the inverted-gamma distribution, may be applied. If we wished to eliminate  $\sigma_{e_i}^2$  from the evidence result of (4.26) then one prior structure which leads to analytic results is to reparameterise the prior covariance matrix for  $\boldsymbol{\theta}_i$  as  $\mathbf{C}'_{\boldsymbol{\theta}_i} = \mathbf{C}_{\boldsymbol{\theta}_i} / \sigma_{e_i}^2$  in order to eliminate the dependency of  $\boldsymbol{\Theta}$  and  $\Phi$  on  $\sigma_{e_i}^2$ . Such a reparameterisation becomes unnecessary if a uniform prior is assumed for  $\boldsymbol{\theta}_i$ , in which case  $\mathbf{C}_{\boldsymbol{\theta}_i}^{-1} \rightarrow \mathbf{0}$  and  $\boldsymbol{\Theta}$  and  $\Phi$  no longer depend upon  $\sigma_{e_i}^2$ .

#### 4.1.4.2 Determination of the MAP state estimate

Having obtained expressions for the state evidence and assigned the prior probabilities  $P_{s_i}$  the posterior state probability (4.18) may then be evaluated for all  $i$  in order to identify the MAP state estimate. For large numbers of states  $N_s$  this can be highly computationally intensive, in which case some sub-optimal search procedure must be devised. This issue is addressed in later chapters for Bayesian click removal.

#### 4.1.5 Sequential Bayesian classification

Sequential classification is concerned with updating classification estimates as new data elements are input. Typically  $\mathbf{x}$  will contain successive data

samples from a time series, and an example is a real-time signal processing application in which new samples are received from the measurement system one-by-one and we wish to refine some classification of the data as more data is received. Say we have observed  $k$  samples which are placed in vector  $\mathbf{x}_k$ . The decision theory of previous sections can be used to calculate the posterior state probability  $p(s_i | \mathbf{x}_k)$  for each state  $s_i$ . The sequential classification problem can then be stated as follows: given some state estimate for the first  $k$  data samples of  $\mathbf{x}_k$  a sequence, form a revised state estimate when a new data sample  $x_{k+1}$  is observed.

The optimal Bayes decision procedure requires minimisation of the conditional risk (4.19) and hence evaluation of all the state posterior probabilities  $p(s_i | \mathbf{x}_k)$ . Considering (4.18), the prior state probabilities  $p(s_i)$  do not change as more data is received, and hence the problem is reduced to that of updating the evidence term  $p(\mathbf{x} | s_i)$  for each possible state with each incoming data sample. This update is specified in recursive form:

$$p(\mathbf{x}_{k+1} | s_i) = g(p(\mathbf{x}_k | s_i), x_{k+1}) \quad (4.31)$$

where  $g(\cdot)$  is the sequential updating function. Simple application of the product rule for probabilities gives the required update in terms of the distribution  $p(x_{k+1} | \mathbf{x}_k, s_i)$ :

$$p(\mathbf{x}_{k+1} | s_i) = p(x_{k+1} | \mathbf{x}_k, s_i) p(\mathbf{x}_k | s_i). \quad (4.32)$$

In some simple cases it is possible to write down this distribution directly from modelling considerations (for example in classification for known fixed parameter linear models). For the general multivariate Gaussian distribution with known mean vector  $\mathbf{m}_\mathbf{x}$  and covariance matrix  $\mathbf{C}_\mathbf{x}$ , the required update can be derived by considering the manner in which  $\mathbf{C}_\mathbf{x}$  and its inverse are modified by the addition of a new element to  $\mathbf{x}$  (see [173, pp.157-160]).

Here we consider the linear Gaussian model with unknown parameters (4.23) which leads to an evidence expression with the form (4.26). The simplest update is obtained by consideration of how the term  $\theta_i^{\text{MAP}}$ , the MAP parameter estimate for a particular state or model, is modified by the input of new data point  $x_{k+1}$ . The ‘ $i$ ’ notation indicating a particular state  $s_i$  is dropped from now on since the update for each state  $s_i$  can be treated as a separate problem. Subscripts now refer to the number of data samples (i.e.  $k$  or  $(k+1)$ ).

For a given model it is assumed that the basis matrix  $\mathbf{G}_k$  corresponding to data  $\mathbf{x}_k$  is updated to  $\mathbf{G}_{k+1}$  with the input of  $x_{k+1}$  by the addition of a new row  $\mathbf{g}_k^T$ :

$$\mathbf{G}_{k+1} = \begin{bmatrix} \mathbf{G}_k \\ - \\ \mathbf{g}_k^T \end{bmatrix} \quad (4.33)$$

Note that even though  $\mathbf{g}_k$  is associated with time instant  $k + 1$  we write it as  $\mathbf{g}_k$  for notational simplicity.

The MAP parameter estimate with  $k$  samples of input is given by  $\boldsymbol{\theta}_k^{\text{MAP}} = \boldsymbol{\Phi}_k^{-1} \boldsymbol{\Theta}_k$  (4.27). The first term in  $\boldsymbol{\Phi}_k$  (4.28) is  $\mathbf{G}_k^T \mathbf{G}_k$ . A recursive update for this term is obtained by direct multiplication using (4.33):

$$\mathbf{G}_{k+1}^T \mathbf{G}_{k+1} = \mathbf{G}_k^T \mathbf{G}_k + \mathbf{g}_k \mathbf{g}_k^T. \quad (4.34)$$

The second term of  $\boldsymbol{\Phi}_k$  (4.28) is  $\sigma_e^2 \mathbf{C}_\theta^{-1}$ . This remains unchanged with the input of  $x_{k+1}$  since it contains parameters of the prior modelling assumptions which are independent of the input data samples. The recursive update for  $\boldsymbol{\Phi}_k$  is thus given by

$$\boldsymbol{\Phi}_{k+1} = \boldsymbol{\Phi}_k + \mathbf{g}_k \mathbf{g}_k^T. \quad (4.35)$$

Now consider updating  $\boldsymbol{\Theta}_k$  (4.29). The first term is  $\mathbf{G}_k^T \mathbf{x}_k$ . Clearly  $\mathbf{x}_{k+1}$  is given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_k \\ - \\ x_{k+1} \end{bmatrix}, \quad (4.36)$$

and hence by direct multiplication:

$$\mathbf{G}_{k+1}^T \mathbf{x}_{k+1} = \mathbf{G}_k^T \mathbf{x}_k + \mathbf{g}_k x_{k+1}. \quad (4.37)$$

As for  $\boldsymbol{\Phi}_k$ , the second term of  $\boldsymbol{\Theta}_k$  (4.29) is unchanged with the input of  $x_{k+1}$ , so the recursive update for  $\boldsymbol{\Theta}$  is

$$\boldsymbol{\Theta}_{k+1} = \boldsymbol{\Theta}_k + \mathbf{g}_k x_{k+1}. \quad (4.38)$$

The updates derived for  $\boldsymbol{\Phi}_k$  (4.35) and  $\boldsymbol{\Theta}_k$  (4.38) are now in the form required for the extended RLS formulae given in appendix B. Use of these formulae, including the determinant updating formulae for  $|\boldsymbol{\Phi}_k|$  give the following update for the evidence (note that  $\mathbf{P} = \boldsymbol{\Phi}^{-1}$ ):

$$\begin{aligned} \mathbf{k}_{k+1} &= \frac{\mathbf{P}_k \mathbf{g}_k}{1 + \mathbf{g}_k^T \mathbf{P}_k \mathbf{g}_k} \\ \alpha_{k+1} &= x_{k+1} - \boldsymbol{\theta}_k^{\text{MAP}T} \mathbf{g}_k \\ \boldsymbol{\theta}_{k+1}^{\text{MAP}} &= \boldsymbol{\theta}_k^{\text{MAP}} + \mathbf{k}_{k+1} \alpha_{k+1} \\ \mathbf{P}_{k+1} &= \mathbf{P}_k - \mathbf{k}_{k+1} \mathbf{g}_k^T \mathbf{P}_k \end{aligned}$$

$$p(\mathbf{x}_{k+1} | s) = p(\mathbf{x}_k | s) \frac{1}{(2\pi\sigma_e^2)^{1/2}} \frac{1}{(1 + \mathbf{g}_k^T \mathbf{P}_k \mathbf{g}_k)^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \alpha_{k+1} \left(x_{k+1} - \boldsymbol{\theta}_{k+1}^{\text{MAP}T} \mathbf{g}_k\right)\right) \quad (4.39)$$

where of course all of the terms are implicitly dependent upon the particular state  $s$ . This update requires at each stage storage of the inverse matrix  $\mathbf{P}_k$ , the MAP parameter estimate  $\boldsymbol{\theta}_k^{\text{MAP}}$  and the state probability  $p(\mathbf{x}_k \mid s_i)$ . Computation of the update requires  $\mathcal{O}(P^2)$  multiplications and additions (recall that  $P$  is the number of unknown parameters in the current state or model), although significant savings can be made in special cases where  $\mathbf{g}_k$  contains zero-valued elements. The MAP parameter estimates are obtained as a by-product of the update scheme, and this will be used in cases where joint estimation of state or model and parameters is required.

The sequential algorithms developed later in the Bayesian de-clicking chapters will be similar to the form summarised here. They not only allow for reasonably efficient real-time operation but will also help in the development of sub-optimal classification algorithms for the cases where the number of states is too large for evaluation of the evidence for all states  $s_i$ .

## 4.2 Signal modelling

The general approach of this book for solving problems in audio is the *model based* approach, in which an attempt is made to formulate a statistical model for the data generation process. The model chosen is considered as part of the prior information about the problem in a similar way to the prior distribution assumed in Bayesian methods. When the model formulated is an accurate representation of the data we can expect to be rewarded with improved processing performance. We have already seen how to deal with statistical model uncertainty between a set of candidate models (or system ‘states’) within a Bayesian framework earlier in this chapter. The general linear model has been used as an example until now. Such a model applies to a time series when the data can be thought of as being composed of a finite number of basis functions (in additive noise). More typically, however, we may wish to include random inputs into the model. One parametric model which achieves this is the ARMA (autoregressive moving-average) model, in which the observed data are generated through the following general linear difference equation:

$$x_n = \sum_{i=1}^P a_i x_{n-i} + \sum_{j=0}^Q b_j e_{n-j} \quad (4.40)$$

*ARMA model*

The transfer function for this model is

$$H(z) = \frac{B(z)}{A(z)}$$

where  $B(z) = \sum_{j=0}^Q b_j z^{-j}$  and  $A(z) = 1 - \sum_{i=1}^P a_i z^{-i}$ .

The model can be seen to consist of applying an IIR filter (see 2.5.1) to the ‘excitation’ or ‘innovation’ sequence  $\{e_n\}$ , which is i.i.d. noise. Generalisations to the model could include the addition of additional deterministic input signals (the ARMAX model [114, 21]) or the inclusion of linear basis functions in the same way as for the general linear model:

$$\mathbf{y} = \mathbf{x} + \mathbf{G}\boldsymbol{\theta}$$

An important special case of the ARMA model is the autoregressive (AR) or ‘all-pole’ (since the transfer function has poles only) model in which  $B(z) = 1$ . This model is used considerably throughout the text and is considered in the next section.

### 4.3 Autoregressive (AR) modelling

A time series model which is fundamental to much of the work in this book is the autoregressive (AR) model, in which the data is modelled as the output of an all-pole filter excited by white noise. This model formulation is a special case of the innovations representation for a stationary random signal in which the signal  $\{X_n\}$  is modelled as the output of a linear time-invariant filter driven by white noise. In the AR case the filtering operation is restricted to a weighted sum of past output values and a white noise innovations input  $\{e_n\}$ :

$$x_n = \sum_{i=1}^P a_i x_{n-i} + e_n. \quad (4.41)$$

The coefficients  $\{a_i; i = 1 \dots P\}$  are the filter coefficients of the all-pole filter, henceforth referred to as the AR parameters, and  $P$ , the number of coefficients, is the order of the AR process. The AR model formulation is closely related to the linear prediction framework used in many fields of signal processing (see e.g. [174, 119]). AR modelling has some very useful properties as will be seen later and these will often lead to simple analytical results where a more general model such as the ARMA model (see previous section) does not. In addition, the AR model has a reasonable basis as a source-filter model for the physical sound production process in many speech and audio signals [156, 187].



### 4.3.1 Statistical modelling and estimation of AR models

If the probability distribution function  $p_e(e_n)$  for the innovation process is known, it is possible to incorporate the AR process into a statistical framework for classification and estimation problems. A straightforward change of variable  $x_n$  to  $e_n$  gives us the distribution for  $x_n$  conditional on the previous  $P$  data values as

$$p(x_n | x_{n-1}, x_{n-2}, \dots, x_{n-P}) = p_e(x_n - \sum_{i=1}^P a_i x_{n-i}) \quad (4.42)$$

Since the excitation sequence is i.i.d. we can write the joint probability for a contiguous block of  $N - P$  data samples  $x_{P+1} \dots x_N$  conditional upon the first  $P$  samples  $x_1 \dots x_P$  as

$$p(x_{P+1}, x_{P+2}, \dots, x_N | x_1, x_2, \dots, x_P) = \prod_{n=P+1}^N p_e(x_n - \sum_{i=1}^P a_i x_{n-i}) \quad (4.43)$$

This is now expressed in matrix-vector notation. The data samples  $x_1, \dots, x_N$  and parameters  $a_1, a_2, \dots, a_{P-1}, a_P$  are written as column vectors of length  $N$  and  $P$ , respectively:

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T, \quad \mathbf{a} = [a_1 \ a_2 \ \dots \ a_{P-1} \ a_P]^T \quad (4.44)$$

$\mathbf{x}$  is partitioned into  $\mathbf{x}_0$ , which contains the first  $P$  samples  $x_1, \dots, x_P$ , and  $\mathbf{x}_1$  which contains the remaining  $(N - P)$  samples  $x_{P+1} \dots x_N$ :

$$\mathbf{x}_0 = [x_1 \ x_2 \ \dots \ x_P]^T, \quad \mathbf{x}_1 = [x_{P+1} \ \dots \ x_N]^T \quad (4.45)$$

The AR modelling equation of (4.41) is now rewritten for the block of  $N$  data samples as

$$\mathbf{x}_1 = \mathbf{G} \mathbf{a} + \mathbf{e} \quad (4.46)$$

where  $\mathbf{e}$  is the vector of  $(N - P)$  excitation values and the  $((N - P) \times P)$  matrix  $\mathbf{G}$  is given by

$$\mathbf{G} = \begin{bmatrix} x_P & x_{P-1} & \cdots & x_2 & x_1 \\ x_{P+1} & x_P & \cdots & x_3 & x_2 \\ \vdots & & \ddots & & \vdots \\ x_{N-2} & x_{N-3} & \cdots & x_{N-P} & x_{N-P-1} \\ x_{N-1} & x_{N-2} & \cdots & x_{N-P+1} & x_{N-P} \end{bmatrix} \quad (4.47)$$

The conditional probability expression (4.43) now becomes

$$p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{a}) = p_e(\mathbf{x}_1 - \mathbf{G}\mathbf{a}) \quad (4.48)$$

and in the case of a zero-mean Gaussian excitation we obtain

$$p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{G}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{G}\mathbf{a})\right) \quad (4.49)$$

Note that this introduces a variance parameter  $\sigma_e^2$  which is in general unknown. The PDF given is thus implicitly conditional on  $\sigma_e^2$  as well as  $\mathbf{a}$  and  $\mathbf{x}_0$ .

The form of the modelling equation of (4.46) looks identical to that of the general linear parametric model used to illustrate previous sections (4.1). We have to bear in mind, however, that  $\mathbf{G}$  here depends upon the data values themselves, which is reflected in the conditioning of the distribution of  $\mathbf{x}_1$  upon  $\mathbf{x}_0$ . It can be argued that this conditioning becomes an insignificant ‘end-effect’ for  $N \gg P$  [155] and we can then make an approximation to obtain the likelihood for  $\mathbf{x}$ :

$$p(\mathbf{x}|\mathbf{a}) \approx p(\mathbf{x}_1|\mathbf{a}, \mathbf{x}_0), \quad N \gg P \quad (4.50)$$

How much greater than  $P$   $N$  must be will in fact depend upon the pole positions of the AR process. Using this result an approximate ML estimator for  $\mathbf{a}$  can be obtained by maximisation w.r.t.  $\mathbf{a}$ , from which we obtain the well-known *covariance* estimate for the AR parameters,

$$\mathbf{a}^{\text{Cov}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{x}_1 \quad (4.51)$$

which is equivalent to a minimisation of the sum-squared prediction error over the block,  $E = \sum_{i=P+1}^N e_i^2$ , and has the same form as the ML parameter estimate in the general linear model.

Consider now an alternative form for the vector model equation (4.46) which will be used in subsequent work for Bayesian detection of clicks and interpolation of AR data:

$$\mathbf{e} = \mathbf{A}\mathbf{x} \quad (4.52)$$

where  $\mathbf{A}$  is the  $((N-P) \times (N))$  matrix defined as

$$\mathbf{A} = \begin{bmatrix} -a_P & \cdots & -a_1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -a_P & \cdots & -a_1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \cdots & 0 & 0 & -a_P & \cdots & -a_1 & 1 & 0 & 0 \\ 0 & \cdots & 0 & 0 & -a_P & \cdots & -a_1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & -a_P & \cdots & -a_1 & 1 \end{bmatrix} \quad (4.53)$$

The conditional likelihood for white Gaussian excitation is then rewritten as:

$$p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}\right) \quad (4.54)$$

In order to obtain the exact (i.e. not conditional upon  $\mathbf{x}_0$ ) likelihood we need the distribution  $p(\mathbf{x}_0|\mathbf{a})$ , since

$$p(\mathbf{x}|\mathbf{a}) = p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a})p(\mathbf{x}_0|\mathbf{a})$$

In appendix C this additional term is derived, and the exact likelihood for all elements of  $\mathbf{x}$  is shown to require only a simple modification to the conditional likelihood, giving:

$$p(\mathbf{x} | \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N}{2}} |\mathbf{M}_{\mathbf{x}_0}|^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{M}_{\mathbf{x}}^{-1} \mathbf{x}\right) \quad (4.55)$$

where

$$\mathbf{M}_{\mathbf{x}}^{-1} = \mathbf{A}^T \mathbf{A} + \begin{bmatrix} \mathbf{M}_{\mathbf{x}_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.56)$$

and  $\mathbf{M}_{\mathbf{x}_0}$  is the autocovariance matrix for  $P$  samples of data drawn from AR process  $\mathbf{a}$  with unit variance excitation. Note that this result relies on the assumption of a *stable* AR process. As seen in the appendix,  $\mathbf{M}_{\mathbf{x}_0}^{-1}$  is straightforwardly obtained in terms of the AR coefficients for any given stable AR model  $\mathbf{a}$ . In problems where the AR parameters are known beforehand but certain data elements are unknown or missing, as in click removal or interpolation problems, it is thus simple to incorporate the true likelihood function in calculations. In practice it will often not be necessary to use the exact likelihood since it will be reasonable to fix at least  $P$  ‘known’ data samples at the start of any data block. In this case the conditional likelihood (4.54) is the required quantity. Where  $P$  samples cannot be fixed it will be necessary to use the exact likelihood expression (4.55) as the conditional likelihood will perform badly in estimating missing data points within  $\mathbf{x}_0$ .

While the exact likelihood is quite easy to incorporate in missing data or interpolation problems with known  $\mathbf{a}$ , it is much more difficult to use for AR parameter estimation since the functions to maximise are non-linear in the parameters  $\mathbf{a}$ . Hence the linearising approximation of equation (4.50) will usually be adopted for the likelihood when the parameters are unknown.

In this section we have shown how to calculate exact and approximate likelihoods for AR data, in two different forms: one as a quadratic form in the data  $\mathbf{x}$  and another as a quadratic (or approximately quadratic) form in the parameters  $\mathbf{a}$ . This likelihood will appear on many subsequent occasions throughout the book.

#### 4.4 State-space models, sequential estimation and the Kalman filter

Most of the models encountered in this book, including the AR and ARMA models, can be expressed in *state space* form:

$$y_n = Z\alpha_n + v_n \quad (\text{Observation equation}) \quad (4.57)$$

$$\alpha_{n+1} = T\alpha_n + He_n \quad (\text{State update equation}) \quad (4.58)$$

In the top line, the *observation equation*, the observed data  $y_n$  is expressed in terms of an unobserved state  $\alpha_n$  and a noise term  $v_n$ .  $v_n$  is uncorrelated (i.e.  $E[v_n v_m^T] = 0$  for  $n \neq m$ ) and zero mean, with covariance  $C_v$ . In the second line, the *state update equation*, the state  $\alpha_n$  is updated to its new value  $\alpha_n$  at time  $n+1$  and a second noise term  $e_n$ .  $e_n$  is uncorrelated (i.e.  $E[e_n e_m^T] = 0$  for  $n \neq m$ ) and zero mean, with covariance  $C_e$ , and is also uncorrelated with  $v_n$ . Note that in general the state  $\alpha_n$ , observation  $y_n$  and noise terms  $e_n$  /  $v_n$  can be column vectors and the constants  $Z$ ,  $T$  and  $H$  are then matrices of the implied dimensionality. Also note that all of these constants can be made time index dependent without altering the form of the results given below.

Take, for example, an AR model  $\{x_t\}$  observed in noise  $\{v_n\}$ , so that the equations in standard form are:

$$y_n = x_n + v_n$$

$$x_n = \sum_{i=1}^P a_i x_{n-i} + e_n$$

One way to express this in state-space form is as follows:

$$\alpha_n = [x_n \quad x_{n-1} \quad x_{n-2} \quad \dots \quad x_{n-P+1}]^T$$

$$T = \begin{bmatrix} a_1 & a_2 & \dots & a_P \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$H = [1 \quad 0 \quad \dots \quad 0]^T$$

$$Z = H^T$$

$$C_e = \sigma_e^2$$

$$C_v = \sigma_v^2$$

The state-space form is useful since some elegant results exist for the general form which can be applied in many different special cases. In par-

ticular, we will use it for sequential estimation of the state  $\alpha_n$ . In probabilistic terms this will involve updating the posterior probability for  $\alpha_n$  with the input of a new observation  $y_{n+1}$ :

$$p(\alpha_{n+1}|\mathbf{y}_{n+1}) = g(p(\alpha_n|\mathbf{y}_n), y_{n+1})$$

where  $\mathbf{y}_n = [y_1, \dots, y_n]^T$  and  $g(\cdot)$  denotes the sequential updating function.

Suppose that the noise sources  $e_n$  and  $v_n$  are Gaussian. Assume also that an initial state probability or prior  $p(\alpha_0)$  exists and is Gaussian  $N(m_0, C_0)$ . Then the posterior distributions are all Gaussian themselves and the posterior distribution for  $\alpha_n$  is fully represented by its *sufficient statistics*: its mean  $a_n = E[\alpha_n|y_0, \dots, y_n]$  and covariance matrix  $P_n = E[(\alpha_n - a_n)(\alpha_n - a_n)^T | y_0, \dots, y_n]$ . The *Kalman* filter [100, 5, 91] performs the update efficiently, as follows:

1. Initialise:  $a_0 = m_0, P_0 = C_0$

2. Repeat for  $n = 1$  to  $N$ :

(a) **Prediction:**

$$\begin{aligned} a_{n|n-1} &= T a_{n-1} \\ P_{n|n-1} &= T P_{n-1} T^T + H C_e H^T \end{aligned}$$

(b) **Correction:**

$$\begin{aligned} a_n &= a_{n|n-1} + K_n (y_n - Z a_{n|n-1}) \\ P_n &= (I - K_n Z) P_{n|n-1} \end{aligned}$$

where

$$K_n = P_{n|n-1} Z^T (Z P_{n|n-1} Z^T + C_v)^{-1} \quad (\text{Kalman Gain})$$

Here  $a_{n|n-1}$  is the predictive mean  $E[\alpha_n|\mathbf{y}_{n-1}]$ ,  $P_{n|n-1}$  the predictive covariance  $E[(\alpha_n - a_{n|n-1})(\alpha_n - a_{n|n-1})^T | \mathbf{y}_{n-1}]$  and  $I$  denotes the (appropriately sized) identity matrix.

We have thus far given a purely probabilistic interpretation of the Kalman filter for normally distributed noise sources and initial state. A more general interpretation is available [100, 5, 91]: the Kalman filter gives the best possible linear estimator for the state in a mean-squared error sense (MSE), whatever the probability distributions.

#### 4.4.1 The prediction error decomposition

One remarkable property of the Kalman filter is the *prediction error decomposition* which allows exact sequential evaluation of the *likelihood* function

for the observations. If we suppose that the model depends upon some hyperparameters  $\theta$ , then the Kalman filter updates sequentially, for a particular value of  $\theta$ , the density  $p(\alpha_n | \mathbf{y}_n, \theta)$ . We define the likelihood for  $\theta$  in this context to be:

$$p(\mathbf{y}_n | \theta) = \int p(\alpha_n, \mathbf{y}_n | \theta) d\alpha_n$$

from which the ML or MAP estimator for  $\theta$  can be obtained by optimisation. The prediction error decomposition [91, pp.125-126] calculates this term from the outputs of the Kalman filter:

$$\log(p(\mathbf{y}_n | \theta)) = -\frac{Mn}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^n \log |F_t| - \frac{1}{2} \sum_{t=1}^n w_t^T F_t^{-1} w_t \quad (4.59)$$

where

$$\begin{aligned} F_t &= Z P_{t|t-1} Z^T + C_v \\ w_t &= y_t - Z a_{t|t-1} \end{aligned}$$

and where  $M$  is the dimension of the observation vector  $y_n$ .

#### 4.4.2 Relationships with other sequential schemes

The Kalman filter is closely related to other well known schemes such as recursive least squares (RLS) [94], which can be obtained as a special case of the Kalman filter when  $T = I$ , the identity matrix, and  $C_e = 0$  (i.e. a system with non-dynamic states - an example of this would be the general linear model used earlier in the chapter, using a time dependent observation vector  $Z_n = \mathbf{g}_n^T$ ).

### 4.5 Expectation-maximisation (EM) for MAP estimation

The Expectation-maximisation (EM) algorithm [43] is an iterative procedure for finding modes of a posterior distribution or likelihood function, particularly in the context of ‘missing data’. EM has been used quite extensively in the signal processing literature for maximum likelihood parameter estimation, see e.g. [59, 195, 136, 168]. Within the audio field Veldhuis [192, appendix E.2] derives the EM algorithm for AR parameter estimation in the presence of missing data, which is closely related to the audio interpolation problems of subsequent chapters. The notation used here is essentially similar to that of Tanner [172, pp.38-57].

The problem is formulated in terms of observed data  $\mathbf{y}$ , parameters  $\boldsymbol{\theta}$  and unobserved (‘latent’ or ‘missing’) data  $\mathbf{x}$ . EM will be useful in certain cases where it is straightforward to manipulate the conditional posterior distributions  $p(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$  and  $p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ , but perhaps not straightforward to deal with the marginal distributions  $p(\boldsymbol{\theta}|\mathbf{y})$  and  $p(\mathbf{x}|\mathbf{y})$ . This will be the case in many of the model-based interpolation and estimation schemes of subsequent chapters. We will typically wish to estimate missing or noisy data  $\mathbf{x}$  in the audio restoration tasks of this book, so we consider a slightly unconventional formulation of EM which treats  $\boldsymbol{\theta}$  as ‘nuisance’ parameters and forms the MAP estimate for  $\mathbf{x}$ :

$$\mathbf{x}^{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}} \{p(\mathbf{x}|\mathbf{y})\} = \underset{\mathbf{x}}{\operatorname{argmax}} \left\{ \int_{\boldsymbol{\theta}} p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta} \right\}$$

The basic EM algorithm can be summarised as:

**1. Expectation step:**

Given the current estimate  $\mathbf{x}^i$ , calculate:

$$\begin{aligned} Q(\mathbf{x}, \mathbf{x}^i) &= \int_{\boldsymbol{\theta}} \log(p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})) p(\boldsymbol{\theta}|\mathbf{x}^i, \mathbf{y}) d\boldsymbol{\theta} \\ &= \mathbb{E}[\log(p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})) | \mathbf{x}^i, \mathbf{y}] \end{aligned} \quad (4.60)$$

**2. Maximisation step:**

$$\mathbf{x}^{i+1} = \underset{\mathbf{x}}{\operatorname{argmax}} \{Q(\mathbf{x}, \mathbf{x}^i)\} \quad (4.61)$$

These two steps are iterated until convergence is achieved. The algorithm is guaranteed to converge to a stationary point of  $p(\mathbf{x}|\mathbf{y})$ , although we must beware of convergence to local maxima when the posterior distribution is multimodal. The starting point  $\mathbf{x}^0$  determines which posterior mode is reached and can be critical in difficult applications.

## 4.6 Markov chain Monte Carlo (MCMC)

Rapid increases in available computational power over the last few years have led to a revival of interest in Markov chain Monte Carlo (MCMC) simulation methods [131, 92]. The object of these methods is to draw *samples* from some target distribution  $\pi(\omega)$  which may be too complex for direct estimation procedures. The MCMC approach sets up an irreducible, aperiodic Markov chain whose stationary distribution is the target distribution of interest,  $\pi(\omega)$ . The Markov chain is then simulated from some arbitrary starting point and convergence in distribution to  $\pi(\omega)$  is then guaranteed

under mild conditions as the number of state transitions (iterations) approaches infinity [175]. Once convergence is achieved, subsequent samples from the chain form a (dependent) set of samples from the target distribution, from which Monte Carlo estimates of desired quantities related to the distribution may be calculated.

For the statistical reconstruction tasks considered in this book, the target distribution will be the *joint* posterior distribution for all unknowns,  $p(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y})$ , from which samples of the unknowns  $\mathbf{x}$  and  $\boldsymbol{\theta}$  will be drawn conditional upon the observed data  $\mathbf{y}$ . Since the joint distribution can be factorised as  $p(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) = p(\boldsymbol{\theta} | \mathbf{x}, \mathbf{y})p(\mathbf{x} | \mathbf{y})$  it is clear that the samples in  $\mathbf{x}$  which are extracted from the joint distribution are equivalent to samples from the *marginal* posterior  $p(\mathbf{x} | \mathbf{y})$ . The sampling method thus implicitly performs the (generally) analytically intractable marginalisation integral w.r.t.  $\boldsymbol{\theta}$ .

The Gibbs Sampler [64, 63] is perhaps the most popular form of MCMC currently in use for the exploration of posterior distributions. This method, which can be derived as a special case of the more general Metropolis-Hastings method [92], requires the full specification of conditional posterior distributions for each unknown parameter or variable. Suppose that the reconstructed data and unknown parameters are split into (possibly multivariate) subsets  $\mathbf{x} = \{x_0, x_1, \dots, x_{N-1}\}$  and  $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_q\}$ . Arbitrary starting values  $\mathbf{x}^0$  and  $\boldsymbol{\theta}^0$  are assigned to the unknowns. A single iteration of the Gibbs Sampler then comprises sampling each variable from its conditional posterior with all remaining variables fixed to their current sampled value. The  $(i + 1)$ th iteration of the sampler may be summarised as:

$$\begin{aligned}
\theta_1^{i+1} &\sim p(\theta_1 | \theta_2^i \dots, \theta_q^i, x_0^i, x_1^i, \dots, x_{N-1}^i, \mathbf{y}) \\
\theta_2^{i+1} &\sim p(\theta_2 | \theta_1^{i+1}, \dots, \theta_q^i, x_0^i, x_1^i, \dots, x_{N-1}^i, \mathbf{y}) \\
&\vdots \\
\theta_q^{i+1} &\sim p(\theta_q | \theta_1^{i+1}, \theta_2^{i+1} \dots, x_0^i, x_1^i, \dots, x_{N-1}^i, \mathbf{y}) \\
x_0^{i+1} &\sim p(x_0 | \theta_1^{i+1}, \theta_2^{i+1} \dots, \theta_q^{i+1}, x_1^i, \dots, x_{N-1}^i, \mathbf{y}) \\
x_1^{i+1} &\sim p(x_1 | \theta_1^{i+1}, \theta_2^{i+1} \dots, \theta_q^{i+1}, x_0^{i+1}, x_2^i, \dots, x_{N-1}^i, \mathbf{y}) \\
&\vdots \\
x_{N-1}^{i+1} &\sim p(x_{N-1} | \theta_1^{i+1}, \theta_2^{i+1} \dots, \theta_q^{i+1}, x_0^{i+1}, x_1^{i+1}, \dots, x_{N-2}^i, \mathbf{y})
\end{aligned}$$

where the notation ‘ $\sim$ ’ denotes that the variable to the left is drawn as a random independent sample from the distribution to the right.

The utility of the Gibbs Sampler arises as a result of the fact that the conditional distributions, under appropriate choice of parameter and data subsets ( $\theta_i$  and  $x_j$ ), will be more straightforward to sample than the full posterior. Multivariate parameter and data subsets can be expected to lead



to faster convergence in terms of number of iterations (see e.g. [113, 163]), but there may be a trade-off in the extra computational complexity involved per iteration. Convergence properties are a difficult and important issue and concrete results are as yet few and specialised. However, geometric convergence rates can be found in the important case where the posterior distribution is Gaussian or approximately Gaussian (see [162] and references therein). Numerous (but mostly *ad hoc*) convergence diagnostics have been devised for more general scenarios and a review may be found in [40].

Once the sampler has converged to the desired posterior distribution, inference can easily be made from the resulting samples. One useful means of analysis is to form histograms of any parameters of interest. These converge in the limit to the true marginal posterior distribution for those parameters and can be used to estimate MAP values and Bayesian confidence intervals, for example. Alternatively a Monte Carlo estimate can be made for the expected value of any desired posterior functional  $f(\cdot)$  as a finite summation:

$$E[f(\mathbf{x})|\mathbf{y}] \approx \frac{\sum_{i=N_0+1}^{N_{\max}} f(\mathbf{x}^i)}{N_{\max} - N_0} \quad (4.62)$$

where  $N_0$  is the convergence ('burn in') time and  $N_{\max}$  is the total number of iterations. The MMSE, for example, is simply the posterior mean, estimated by setting  $f(\mathbf{x}) = \mathbf{x}$  in (4.62).

MCMC methods are highly computer-intensive and will only be applicable when off-line processing is acceptable and the problem is sufficiently complex to warrant their sophistication. However, they are currently unparalleled in ability to solve the most challenging of modelling problems.

## 4.7 Conclusions

We have used this chapter to review and develop the principles of parameter estimation and classification with an emphasis on the Bayesian framework. We have considered a linear Gaussian parametric model by way of illustration since the results obtained are closely related to those of some later sections of the book. The autoregressive model which is also fundamental to later chapters has been introduced and discussed within the Bayesian scheme and sequential estimation has been introduced. The Kalman filter was seen to be an appropriate method for implementation of such sequential schemes. Finally some more advanced methods for exploration of a posterior distribution were briefly described: the EM algorithm and MCMC methods, which find application in the last chapter of the book.



## Part II

# Basic Restoration Procedures



# 5

## Removal of Clicks

The term ‘clicks’ is used here to refer to a generic localised type of degradation which is common to many audio media. We will classify all finite duration defects which occur at random positions in the waveform as clicks. Clicks are perceived in a number of ways by the listener, ranging from tiny ‘tick’ noises which can occur in any recording medium, including modern digital sources, through to the characteristic ‘scratch’ and ‘crackle’ noise associated with most analogue disc recording methods. For example, a poor quality 78rpm record might typically have around 2,000 clicks per second of recorded material, with durations ranging from less than  $20\mu\text{s}$  up to 4ms in extreme cases. See figure 5.11 for a typical example of a recorded music waveform degraded by localised clicks. In most examples at least 90% of samples remain undegraded, so it is reasonable to hope that a convincing restoration can be achieved.

There are many mechanisms by which clicks can occur. Typical examples are specks of dirt and dust adhering to the grooves of a gramophone disc or granularity in the material used for pressing such a disc (see figures 1.1-1.3). Further click-type degradation may be caused through damage to the disc in the form of small scratches on the surface. Similar artefacts are encountered in other analogue media, including optical film sound tracks and early wax cylinder recordings, although magnetic tape recordings are generally free of clicks. Ticks can occur in digital recordings as a result of poorly concealed digital errors and timing problems.

Peak-related distortion, occurring as a result either of overload during recording or wear and tear during playback, can give rise to a similar perceived effect to clicks, but is really a different area which should receive

separate attention, even though click removal systems can often go some way towards alleviating the worst effects.

## 5.1 Modelling of clicks

Localised defects may be modelled in many different ways. For example, a defect may be additive to the underlying audio signal, or it may replace the signal altogether for some short period. An additive model has been found to be acceptable for most surface defects in recording media, including small scratches, dust and dirt. A replacement model may be appropriate for very large scratches and breakages which completely obliterate any underlying signal information, although such defects usually excite long-term resonances in mechanical playback systems and must be treated differently (see chapter 7). Here we will consider primarily the additive model, although many of the results are at least robust to replacement noise.

An additive model for localised degradation can be expressed as:

$$y_t = x_t + i_t n_t \quad (5.1)$$

where  $x_t$  is the underlying audio signal,  $n_t$  is a corrupting noise process and  $i_t$  is a 0/1 ‘switching’ process which takes the value 1 only when the localised degradation is present. Clearly the value of  $n_t$  is irrelevant to the output when the switch is in position 0 and should thus be regarded only as a conceptual entity which makes the representation of (5.1) straightforward. The statistics of the switching process  $i_t$  thus govern which samples are degraded, while the statistics of  $n_t$  determine the amplitude characteristics of the corrupting process.

The model is quite general and can account for a wide variety of noise characteristics encountered in audio recordings. It assumes for the moment that there is no continuous background noise degradation at locations where  $i_t = 0$ . It assumes also that the degradation process does not interfere with the timing content of the original signal, as observed in  $y_t$ . This is reasonable for all but very severe degradations, which might temporarily upset the speed of playback, or actual breakages in the medium which have been mechanically repaired (such as a broken disc recording which has been glued back together).

Any procedure which is designed to remove localised defects in audio signals must take account of the typical characteristics of these artefacts. Some important features which are common to many click-degraded audio media include:

- Degradation tends to occur in contiguous ‘bursts’ of corrupted samples, starting at random positions in the waveform and of random duration (typically between 1 and 200 samples at 44.1 kHz sampling

rates). Thus there is strong dependence between successive samples of the switching process  $i_t$ , and the noise cannot be assumed to follow a classical impulsive noise pattern in which single impulses occur independently of each other (the Bernoulli model). It is considerably more difficult to treat clusters of impulsive disturbance than single impulses, since the effects of adjacent impulses can cancel each other in the detection space ('missed detections') or add constructively to give the impression of more impulses ('false alarms').

- The amplitude of the degradation can vary greatly within the same recorded extract, owing to a range of size of physical defects. For example, in many recordings the largest click amplitudes will be well above the largest signal amplitudes, while the smallest audible defects can be more than 40dB below the local signal level (depending on psychoacoustical masking by the signal and the amount of background noise). This leads to a number of difficulties. In particular, large amplitude defects will tend to bias any parameter estimation and threshold determination procedures, leaving smaller defects undetected. As we shall see in section 5.3.1, threshold selection for some detection schemes becomes a difficult problem in this case.

Many approaches are possible for the restoration of such defects. It is clear, however, that the ideal system will process only on those samples which are degraded, leaving the others untouched in the interests of fidelity to the original source. Two tasks can thus be identified for a successful click restoration system. The first is a *detection* procedure in which we estimate the values of the noise switching process  $i_t$ , that is decide which samples are corrupted. The second is an estimation procedure in which we attempt to reconstruct the underlying audio data when corruption is present. A method which assumes that no useful information about the underlying signal is contained in the degraded samples will involve a pure *interpolation* of the audio data based on the value of the surrounding undegraded samples, while more sophisticated techniques will attempt in addition to extract information from the degraded sample values using some degree of explicit noise modelling.

## 5.2 Interpolation of missing samples

At the heart of most click removal methods is an interpolation scheme which replaces missing or corrupted samples with estimates of their true value. It is usually appropriate to assume that clicks have in no way interfered with the timing of the material, so the task is then to fill in the 'gap' with appropriate material of identical duration to the click. As discussed above, this amounts to an interpolation problem which makes use of the

good data values surrounding the corruption and possibly takes account of signal information which is buried in the corrupted sections of data. An effective technique will have the ability to interpolate gap lengths from one sample up to at least 100 samples at a sampling rate of 44.1kHz.

The interpolation problem may be formulated as follows. Consider  $N$  samples of audio data, forming a vector  $\mathbf{x}$ . The corresponding click-degraded data vector is  $\mathbf{y}$ , and the vector of detection values  $i_t$  is  $\mathbf{i}$  (assumed known for the time being). The audio data  $\mathbf{x}$  may be partitioned into two sub-vectors, one containing elements whose value is known (i.e.  $i_t = 0$ ), denoted by  $\mathbf{x}_{-(i)}$ , and the second containing unknown elements which are corrupted by noise ( $i_t = 1$ ), denoted by  $\mathbf{x}_{(i)}$ . Consider, for example, the case where a section of data of length  $l$  samples starting at sample number  $m$  is known to be missing or irrevocably corrupted by noise. The data is first partitioned into three sections: the unknown section  $\mathbf{x}_{(i)} = [x_m, x_{m+1}, \dots, x_{m+l-1}]^T$ , the  $m$  samples to the left of the gap  $\mathbf{x}_{-(i)a} = [x_1, x_2, \dots, x_{m-1}]^T$  and the remaining known samples to the right of the gap  $\mathbf{x}_{-(i)b} = [x_{m+l}, \dots, x_N]^T$ :

$$\mathbf{x} = [\mathbf{x}_{-(i)a}^T \ \mathbf{x}_{(i)}^T \ \mathbf{x}_{-(i)b}^T]^T \quad (5.2)$$

We then form a single vector of known samples  $\mathbf{x}_{-(i)} = [\mathbf{x}_{-(i)a}^T \ \mathbf{x}_{-(i)b}^T]^T$ . For more complicated patterns of missing data a similar but more elaborate procedure is applied to obtain vectors of known and unknown samples. Vectors  $\mathbf{y}$  and  $\mathbf{i}$  are partitioned in a similar fashion. The replacement or interpolation problem requires the estimation of the unknown data  $\mathbf{x}_{(i)}$ , given the observed (corrupted) data  $\mathbf{y}$ . Interpolation will be a statistical estimation procedure for audio signals, which are stochastic in nature, and estimation methods might be chosen to satisfy criteria such as minimum mean-square error (MMSE), maximum likelihood (ML), maximum *a posteriori* (MAP) or some perceptually based criterion.

Numerous methods have been developed for the interpolation of corrupted or missing samples in speech and audio signals. The ‘classical’ approach is perhaps the median filter [184, 151] which can replace corrupted samples with a median value while retaining detail in the signal waveform. A suitable system is described in [101], while a hybrid autoregressive prediction/ median filtering method is presented in [143]. Median filters, however, are too crude to deal with gap lengths greater than a few samples. Other techniques ‘splice’ uncorrupted data from nearby into the gap [115, 152] in such a manner that there is no signal discontinuity at the start or end of the gap. These methods rely on the periodic nature of many speech and music signals and also require a reliable estimate of pitch period.

The most effective and flexible methods to date have been model-based, allowing for the incorporation of reasonable prior information about signal characteristics. A good coverage is given by Veldhuis [192], and a number of interpolators suited to speech and audio signals is presented. These are



based on minimum variance estimation under various modelling assumptions, including sinusoidal, autoregressive and periodic.

We present firstly a general framework for interpolation of signals which can be considered as Gaussian. There then follows a detailed description of some of the more successful techniques which can be considered as special cases of this general framework.

### 5.2.1 *Interpolation for Gaussian signals with known covariance structure*

We derive firstly a general result for the interpolation of zero mean Gaussian signals with known covariance matrix. Many of the interpolators presented in subsequent sections can be regarded as special cases of this interpolator with particular constrained forms for the covariance matrix. The case of non-zero mean signals can be obtained as a straightforward extension of the basic result, but is not presented here as it is not used by the subsequent schemes.

Suppose that a block of  $N$  data samples  $\mathbf{x}$  is partitioned according to known and unknown samples  $\mathbf{x}_{-(i)}$  and  $\mathbf{x}_{(i)}$ . As indicated above, the unknown samples can be specified at arbitrary positions in the block and not necessarily as one contiguous missing chunk. We can express  $\mathbf{x}$  in terms of its known and unknown components as:

$$\mathbf{x} = \mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)}. \quad (5.3)$$

Here  $\mathbf{U}$  and  $\mathbf{K}$  are ‘rearrangement’ matrices which reassemble  $\mathbf{x}$  from the partitions  $\mathbf{x}_{(i)}$  and  $\mathbf{x}_{-(i)}$ .  $\mathbf{U}$  and  $\mathbf{K}$  form a columnwise partition of the  $(N \times N)$  identity matrix:  $\mathbf{U}$  contains all the columns of  $\mathbf{I}$  for which  $i_t = 1$  while  $\mathbf{V}$  contains the remaining columns, for which  $i_t = 0$ .

Then the PDF for the unknown samples conditional upon the known samples is given, using the conditional probability results of chapter 3, by:

$$p(\mathbf{x}_{(i)} | \mathbf{x}_{-(i)}) = \frac{p(\mathbf{x})}{p(\mathbf{x}_{-(i)})} \quad (5.4)$$

$$= \frac{p_{\mathbf{x}}(\mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)})}{p(\mathbf{x}_{-(i)})} \quad (5.5)$$

Given the posterior distribution for the unknown samples we can now design an estimator based upon some appropriate criterion. The MAP interpolation, for example, is then the vector  $\mathbf{x}_{(i)}$  which maximises  $p(\mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)})$  for a given  $\mathbf{x}_{-(i)}$ , since the denominator term in (5.5) is a constant. This posterior probability expression is quite general and applies to any form of PDF for the data vector  $\mathbf{x}$ . For the case of Gaussian random vectors it is well known that the MAP estimator corresponds exactly to the MMSE estimator, hence we derive the MAP estimator as a suitable general purpose interpolation scheme for Gaussian audio signals.

The data block  $\mathbf{x}$  is now assumed to be a random zero mean Gaussian vector with autocorrelation matrix  $\mathbf{R}_{\mathbf{x}} = E[\mathbf{x} \mathbf{x}^T]$  which is strictly positive definite. Then  $p(\mathbf{x})$  is given by the general zero-mean multi-variate Gaussian (A.2) as:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\mathbf{R}_{\mathbf{x}}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{x}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{x}\right) \quad (5.6)$$

Rewriting  $\mathbf{x}$  as in (5.3) gives for two times minus the exponent:

$$\mathbf{x}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{x} = (\mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)})^T \mathbf{R}_{\mathbf{x}}^{-1} (\mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)}) \quad (5.7)$$

Since the exponential function is monotonically increasing in its argument, the vector  $\mathbf{x}_{(i)}$  which minimises this term is the MAP interpolation. We can use the results of vector-matrix differentiation to solve for the minimum, which is given by:

$$\mathbf{x}_{(i)}^{\text{MAP}} = -\mathbf{M}_{(i)(i)}^{-1} \mathbf{M}_{(i)-(i)} \mathbf{x}_{-(i)} \quad (5.8)$$

where

$$\mathbf{M}_{(i)(i)} = \mathbf{U}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{U} \quad \text{and} \quad \mathbf{M}_{(i)-(i)} = \mathbf{U}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{K} \quad (5.9)$$

are submatrices of  $\mathbf{R}_{\mathbf{x}}^{-1}$  defined by the pre- and post-multipliers  $\mathbf{U}$  and  $\mathbf{K}$ . These select particular column and row subsets of  $\mathbf{R}_{\mathbf{x}}^{-1}$  according to which samples of  $\mathbf{x}$  are known or missing. It can readily be shown that this interpolator is also the minimum mean-square error *linear* interpolator for *any* signal vector, Gaussian or non-Gaussian, with known finite autocorrelation matrix.

Thus for a Gaussian signal an interpolation can be made based simply on autocorrelation values up to lag  $N$  or their estimates. Wide-sense stationarity of the process would imply a Toeplitz structure on the matrix. Non-stationarity could in principle be incorporated by allowing  $\mathbf{R}_{\mathbf{x}}$  to be non-Toeplitz, hence allowing for an autocorrelation function which evolves with time in some known fashion, although estimation of the non-stationary autocorrelation function is a separate issue not considered here.

This interpolation scheme can be viewed as a possible practical interpolation method in its own right, allowing for the incorporation of high lag correlations which would not be accurately modelled in all but a very high order AR model, for example. We have obtained some very impressive results with this interpolator in signals with significant long-term correlation structure. The autocorrelation function is estimated from sections of uncorrupted data immediately prior to the missing sections. We do however require the inverse correlation matrix  $\mathbf{R}_{\mathbf{x}}^{-1}$ , which can be a highly computer intensive calculation for large  $N$ . In the stationary case the covariance matrix is Toeplitz and its inverse can be calculated in  $\mathcal{O}(N^2)$  operations

using Trench’s algorithm [86] (which is similar to the Levinson recursion), but computational load may still be too high for many applications if  $N$  is large.

We will see that several of the interpolators of subsequent sections are special cases of this scheme with constrained forms substituted for  $\mathbf{R}_{\mathbf{x}}^{-1}$ . In many of these cases the constrained forms for  $\mathbf{R}_{\mathbf{x}}^{-1}$  will lead to special structures in the solution, such as Toeplitz equations, which can be exploited to give computational efficiency.

#### 5.2.1.1 Incorporating a noise model

We have assumed in the previous section that the corrupted data samples are discarded as missing. However, there may be useful information in the corrupted samples as well, particularly when the click amplitude is very small, which can improve the quality of interpolation compared with the ‘missing data’ interpolator. Re-examining our original model for click-degraded samples,

$$y_t = x_t + i_t n_t \quad (5.10)$$

we see that the interpolation method described so far did not include a model for the additive noise term  $n_t$  and we have discarded all observed samples for which  $i_t = 1$ . This can be remedied by including a model for  $n_t$  which represents the click amplitudes present on a particular recording and performing interpolation conditional upon the complete vector  $\mathbf{y}$  of observed samples. The simplest assumption for the noise model is i.i.d. Gaussian with variance  $\sigma_v^2$ . The basic MAP interpolator is now modified to:

$$\mathbf{x}_{(i)}^{\text{MAP}} = - \left( \mathbf{M}_{(i)(i)} + \frac{1}{\sigma_v^2} \mathbf{I} \right)^{-1} \left( \mathbf{M}_{(i)-(i)} \mathbf{x}_{-(i)} - \frac{1}{\sigma_v^2} \mathbf{y}_{(i)} \right) \quad (5.11)$$

We can see that this version of the interpolator now explicitly uses the values of the corrupted samples  $\mathbf{y}_{(i)}$  in the interpolation. The assumption of an i.i.d. Gaussian noise process for the click amplitudes is rather crude and may lead to problems since the click amplitudes are generally drawn from some heavy-tailed non-Gaussian distribution. Furthermore, we have considered the noise variance  $\sigma_v^2$  as known. However, more realistic non-Gaussian noise modelling can be achieved by the use of iterative techniques which maintain the Gaussian core of the interpolator in a similar form to (5.11) while producing interpolations which are based upon the more realistic heavy-tailed noise distribution. See chapter 12 and [83, 80, 82, 73, 72, 81] for the details of these noise models in the audio restoration problem using Expectation-maximisation (EM) and Markov chain Monte Carlo (MCMC) methods.

Noise models will be considered in more detail in the section on AR model-based interpolators and the chapter on Bayesian click detection.

### 5.2.2 Autoregressive (AR) model-based interpolation

An interpolation procedure which has proved highly successful is the autoregressive (AR) model-based method. This was devised first by Janssen *et al.* [96] for the concealment of uncorrectable errors in CD systems, but was independently derived and applied to the audio restoration problem by Vaseghi and Rayner [190, 187, 191]. As for the general Gaussian case above we present the algorithm in a matrix/vector notation in which the locations of missing samples can be arbitrarily specified within the data block through a detection switching vector  $\mathbf{i}$ . The method can be derived from least-squares (LS) or maximum *a posteriori* (MAP) principles and we present both derivations here, with the least squares derivation first as it has the simplest interpretation.

#### 5.2.2.1 The least squares AR (LSAR) interpolator

Consider first a block of data samples  $\mathbf{x}$  drawn from an AR process with parameters  $\mathbf{a}$ . As detailed in (4.52) and (4.53) we can write the excitation vector  $\mathbf{e}$  in terms of the data vector:

$$\mathbf{e} = \mathbf{A}\mathbf{x} \quad (5.12)$$

$$= \mathbf{A}(\mathbf{U} \mathbf{x}_{(i)} + \mathbf{K} \mathbf{x}_{-(i)}) \quad (5.13)$$

where  $\mathbf{x}$  has been re-expressed in terms of its partition as before. We now define  $\mathbf{A}_{(i)} = \mathbf{A}\mathbf{U}$  and  $\mathbf{A}_{-(i)} = \mathbf{A}\mathbf{K}$ , noting that these correspond to a columnwise partition of  $\mathbf{A}$  corresponding to unknown and known samples, respectively, to give:

$$\mathbf{e} = \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} + \mathbf{A}_{(i)} \mathbf{x}_{(i)} \quad (5.14)$$

The sum squared of the prediction errors over the whole data block is given by:

$$E = \sum_{n=P+1}^N e_n^2 = \mathbf{e}^T \mathbf{e}$$

The least squares (LS) interpolator is now obtained as the interpolated data vector  $\mathbf{x}_{(i)}$  which minimises the sum squared prediction error  $E$ , since  $E$  can be regarded as a measure of the goodness of ‘fit’ of the data to the AR model. In other words, the solution is found as that unknown data vector  $\mathbf{x}_{(i)}$  which minimises  $E$ :

$$\mathbf{x}_{(i)}^{\text{LS}} = \underset{\mathbf{x}_{(i)}}{\operatorname{argmin}}\{E\}.$$

$E$  can be expanded and differentiated using standard vector-matrix calculus to find its minimum:

$$\begin{aligned} E &= \mathbf{e}^T \mathbf{e} \\ \frac{\partial E}{\partial \mathbf{x}_{(i)}} &= 2\mathbf{e}^T \frac{\partial \mathbf{e}}{\partial \mathbf{x}_{(i)}} \\ &= 2(\mathbf{A}_{-(i)} \mathbf{x}_{-(i)} + \mathbf{A}_{(i)} \mathbf{x}_{(i)})^T \mathbf{A}_{(i)} = 0 \end{aligned}$$

Hence, solving for  $\mathbf{x}_{(i)}$ , we obtain:

$$\mathbf{x}_{(i)}^{\text{LS}} = -(\mathbf{A}_{(i)}^T \mathbf{A}_{(i)})^{-1} \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} \quad (5.15)$$

This solution for the missing data involves solution of a set of  $l$  linear equations, where  $l$  is the number of missing samples in the block. When there are at least  $P$  known samples either side of a single contiguous burst of missing samples the LSAR estimate can be efficiently realised in  $\mathcal{O}(l^2)$  multiplies and additions using the Levinson-Durbin recursion [52, 86] since the system of equations is Toeplitz. The computation is further reduced if the banded structure of  $(\mathbf{A}_{(i)}^T \mathbf{A}_{(i)})$  is taken into account. In more general cases where missing samples may occur at arbitrary positions within the data vector it is possible to exploit the Markovian structure of the AR model using the Kalman Filter [5] by expressing the AR model in state space form (see section 5.2.3.5).

#### 5.2.2.2 The MAP AR interpolator

We now derive the maximum *a posteriori* (MAP) interpolator and show that under certain conditions it is identical to the least squares interpolator. If the autoregressive process for the audio data is assumed Gaussian then (5.8) gives the required result. The inverse autocorrelation matrix of the data,  $\mathbf{R}_{\mathbf{x}}^{-1}$ , is required in order to use this result and can be obtained from the results in appendix C as:

$$\mathbf{R}_{\mathbf{x}}^{-1} = \frac{\mathbf{M}_{\mathbf{x}}^{-1}}{\sigma_e^2} = \frac{1}{\sigma_e^2} \left( \mathbf{A}^T \mathbf{A} + \begin{bmatrix} \mathbf{M}_{\mathbf{x}_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right)$$

The required submatrices  $\mathbf{M}_{(i)(i)}$  and  $\mathbf{M}_{(i)-(i)}$  are thus:

$$\mathbf{M}_{(i)(i)} = \mathbf{U}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{U} = \frac{1}{\sigma_e^2} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} + \mathbf{U}^T \begin{bmatrix} \mathbf{M}_{\mathbf{x}_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U} \right)$$

and

$$\mathbf{M}_{(i)-(i)} = \mathbf{U}^T \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{K} = \frac{1}{\sigma_e^2} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} + \mathbf{U}^T \begin{bmatrix} \mathbf{M}_{\mathbf{x}_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{K} \right)$$

Noting that the operation of pre-multiplying a matrix by  $\mathbf{U}^T$  is equivalent to extracting the rows of that matrix which correspond to unknown samples, we can see that the second term in each of these expressions becomes zero when there are no corrupted samples within the first  $P$  samples of the data vector. In this case the least squares and MAP interpolators are exactly equivalent to one another:

$$\mathbf{x}_{(i)}^{\text{MAP}} = \mathbf{x}_{(i)}^{\text{LS}} = -(\mathbf{A}_{(i)}^T \mathbf{A}_{(i)})^{-1} \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)}$$

This equivalence will usually hold in practice, since it will be sensible to choose a data vector which has a reasonably large number of uncorrupted data points before the first missing data point if the estimates are to be reliable. The least squares interpolator is found to perform poorly when missing data occurs in the initial  $P$  samples of  $\mathbf{x}$ , which can be explained in terms of its divergence from the true MAP interpolator under that condition.

### 5.2.2.3 Examples of the LSAR interpolator

See figure 5.1 for examples of interpolation using the LSAR method applied to a music signal. A succession of interpolations has been performed, with increasing numbers of missing samples from left to right in the data (gap lengths increase from 25 samples up to more than 100). The autoregressive model order is 60. The shorter length interpolations are almost indistinguishable from the true signal (left-hand side of figure 5.1(a)), while the interpolation is much poorer as the number of missing samples becomes large (right-hand side of figure 5.1(b)). This is to be expected of any interpolation scheme when the data is drawn from a random process, but the situation can often be improved by use of a higher order autoregressive model. Despite poor accuracy of the interpolant for longer gap lengths, good continuity is maintained at the start and end of the missing data blocks, and the signal appears to have the right ‘character’. Thus effective removal of click artefacts in typical audio sources can usually be achieved.

### 5.2.2.4 The case of unknown AR model parameters

The basic formulation given in (5.15) assumes that the AR parameters are known *a priori*. In practice we may have a robust estimate of the parameters obtained during the detection stage (see section 5.3.1). This, however, is strictly sub-optimal since it will be affected by the values of the corrupted samples and we should perhaps consider interpolation methods which treat the parameters as unknown *a priori*. There are various possible approaches to this problem. The classical time series maximum likelihood method would integrate out the unknown data  $\mathbf{x}_{(i)}$  to give a likelihood function for just the parameters (including the excitation variance), which can then be maximised to give the maximum likelihood solution:

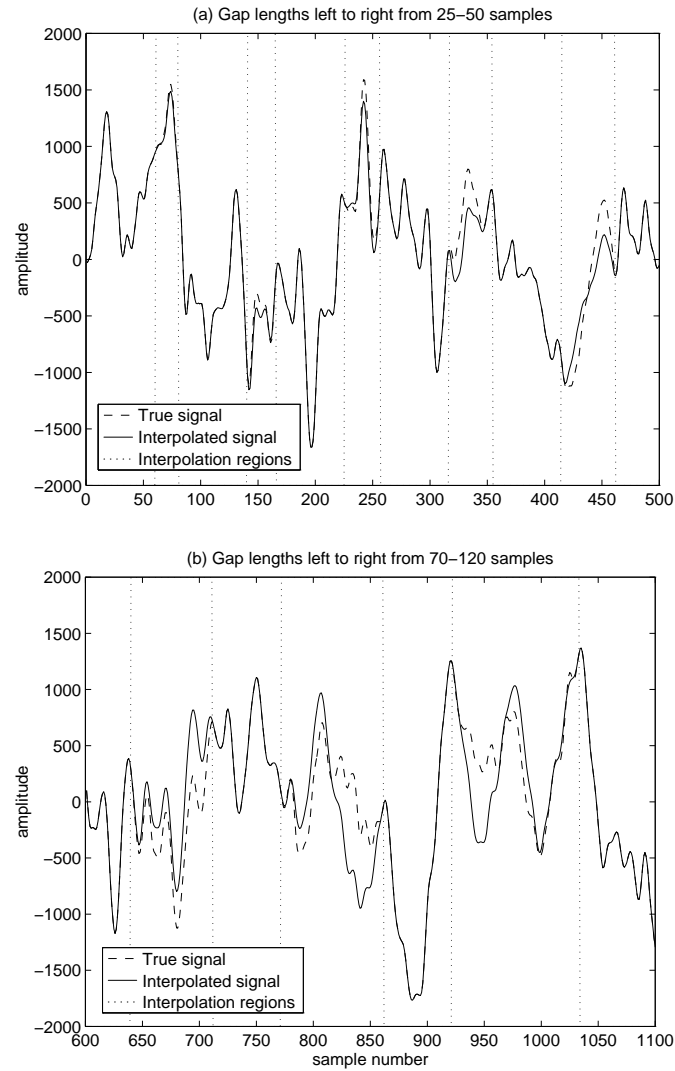


FIGURE 5.1. AR-based interpolation,  $P=60$ , classical chamber music, (a) short gaps, (b) long gaps

$$(\mathbf{a}, \sigma_e^2)^{\text{ML}} = \underset{\mathbf{a}, \sigma_e^2}{\operatorname{argmax}} \left\{ p(\mathbf{x}_{-(i)} | \mathbf{a}, \sigma_e^2) = \int_{\mathbf{x}_{(i)}} p(\mathbf{x} | \mathbf{a}, \sigma_e^2) d\mathbf{x}_{(i)} \right\}.$$

The integral can be performed analytically for the Gaussian case either directly or by use of the *prediction error decomposition* and the Kalman filter (see section 4.4 and [91] for details). The maximisation would be performed iteratively by use of some form of gradient ascent or the expectation-maximisation (EM) algorithm [43] (see section 4.5). A simpler iterative approach, proposed by Janssen, Veldhuis and Vries [96], performs the following steps until convergence is achieved according to some suitable criterion:

1. Initialise  $\mathbf{a}$  and  $\mathbf{x}_{(i)}$
2. Estimate  $\mathbf{a}$  by least squares from  $\mathbf{x}$  with the current estimate of  $\mathbf{x}_{(i)}$  substituted.
3. Estimate  $\mathbf{x}_{(i)}$  by least squares interpolation using the current estimate of  $\mathbf{a}$ .
4. Goto 2.

This simple procedure is quite effective but sensitive to the initialisation and prone sometimes to convergence to local maxima. We note that  $\sigma_e^2$  does not appear as steps 2. and 3. are independent of its value. More sophisticated approaches can be applied which are based upon Bayesian techniques and allow for a prior distribution on the parameters. This solution can again be obtained by numerical techniques, see chapter 12 and [129, 168] for Bayesian interpolation methods using EM and Markov chain Monte Carlo (MCMC) simulation.

In general it will not be necessary to perform an extensive iterative scheme, since a robust initial estimate for the AR parameters can easily be obtained from the raw data, see e.g. [104, 123] for suitable methods. One or two iterations may then be sufficient to give a perceptually high quality restoration.

### 5.2.3 Adaptations to the AR model-based interpolator

The basic AR model-based interpolator performs well in most cases. However, certain classes of signal which do not fit the modelling assumptions (such as periodic pulse-driven voiced speech) and very long gap lengths can lead to an audible ‘dulling’ of the signal or unsatisfactory masking of the original corruption. Increasing the order of the AR model will usually improve the results; however, several developments to the method which are now outlined can lead to improved performance in some scenarios.



### 5.2.3.1 Pitch-based extension to the AR interpolator

Vaseghi and Rayner [191] propose an extended AR model to take account of signals with long-term correlation structure, such as voiced speech, singing or near-periodic music. The model, which is similar to the long term prediction schemes used in some speech coders, introduces extra predictor parameters around the pitch period  $T$ , so that the AR model equation is modified to:

$$x_t = \sum_{i=1}^P x_{n-i} a_i + \sum_{j=-Q}^Q x_{n-T-j} b_j + e_t, \quad (5.16)$$

where  $Q$  is typically smaller than  $P$ . Least squares/ML interpolation using this model is of a similar form to the standard LSAR interpolator, and parameter estimation is straightforwardly derived as an extension of standard AR parameter estimation methods (see section 4.3.1). The method gives a useful extra degree of support from adjacent pitch periods which can only be obtained using very high model orders in the standard AR case. As a result, the ‘under-prediction’ sometimes observed when interpolating long gaps is improved. Of course, an estimate of  $T$  is required, but results are quite robust to errors in this. Veldhuis [192, chapter 4] presents a special case of this interpolation method in which the signal is modelled by one single ‘prediction’ element at the pitch period (i.e.  $Q = 0$  and  $P = 0$  in the above equation).

### 5.2.3.2 Interpolation with an AR + basis function representation

A simple extension of the AR-based interpolator modifies the signal model to include some deterministic basis functions, such as sinusoids or wavelets. Often it will be possible to model most of the signal energy using the deterministic basis, while the AR model captures the correlation structure of the residual. The sinusoid + residual model, for example, has been applied successfully by various researchers, see e.g. [169, 158, 165, 66]. The model for  $x_n$  with AR residual can be written as:

$$x_n = \sum_{i=1}^Q c_i \psi_i[n] + r_n \quad \text{where} \quad r_n = \sum_{i=1}^P a_i r_{n-i} + e_n$$

Here  $\psi_i[n]$  is the  $n$ th element of the  $i$ th basis vector  $\boldsymbol{\psi}_i$  and  $r_n$  is the residual, which is modelled as an AR process in the usual way. For example, with a sinusoidal basis we might take  $\psi_{2i-1}[n] = \cos(\omega_i nT)$  and  $\psi_{2i}[n] = \sin(\omega_i nT)$ , where  $\omega_i$  is the  $i$ th sinusoid frequency. Another simple example of basis functions would be a d.c. offset or polynomial trend. These can be incorporated within exactly the same model and hence the interpolator presented here is a means for dealing also with non-zero mean or smooth underlying trends.

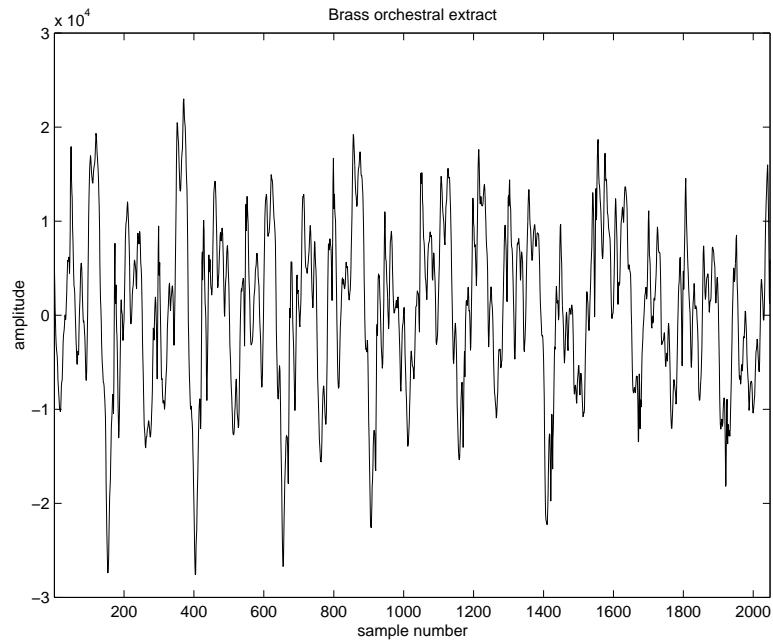


FIGURE 5.2. Original (uncorrupted) audio extract

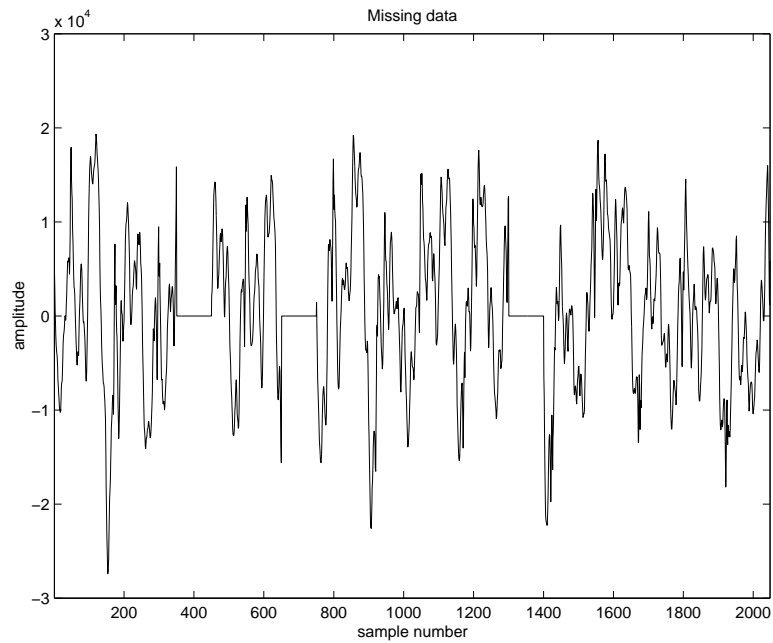


FIGURE 5.3. Audio extract with missing sections (shown as zero amplitude)

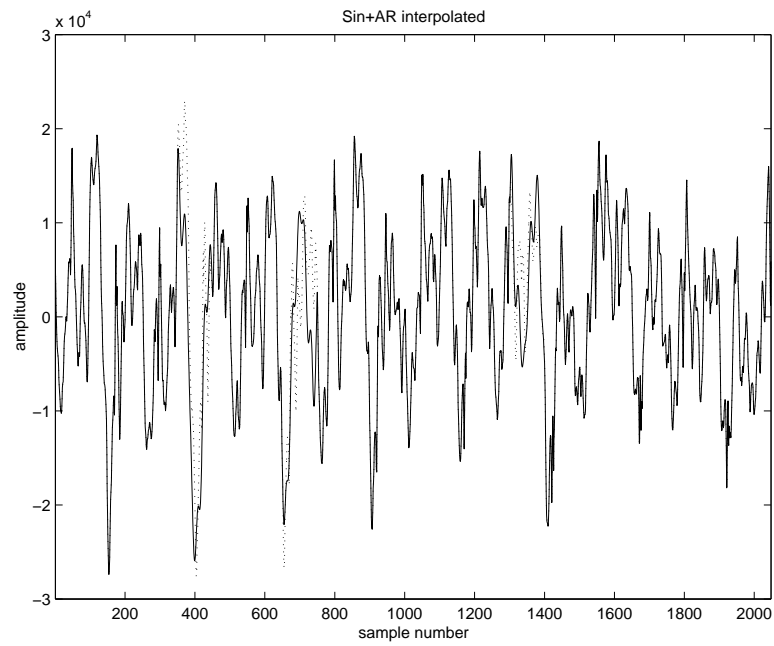


FIGURE 5.4. Sin+AR interpolated audio: dotted line - true original; solid line - interpolated

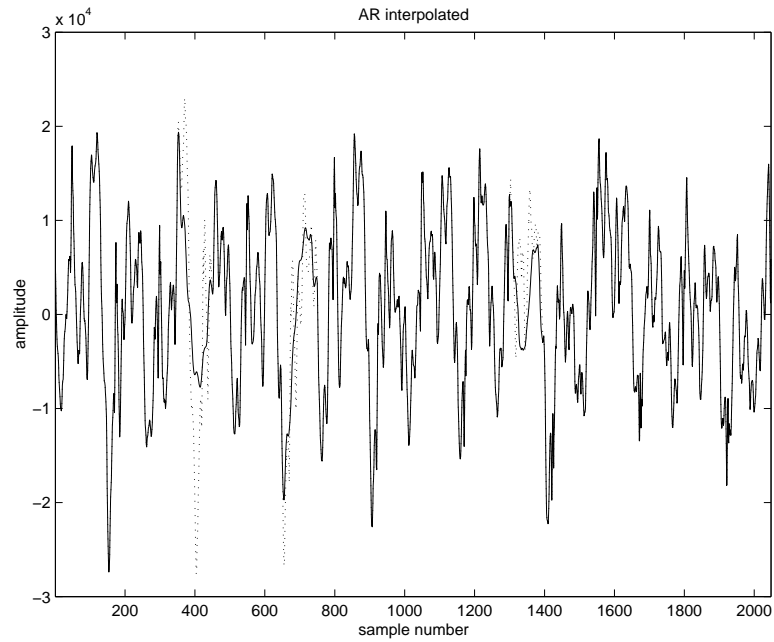


FIGURE 5.5. AR interpolated audio: dotted line - true original; solid line - interpolated

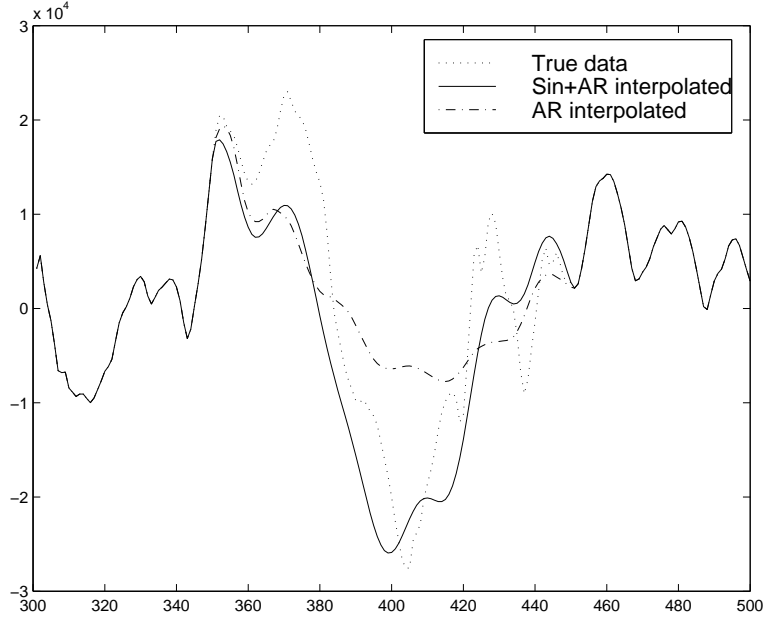


FIGURE 5.6. Comparison of both methods

If we assume for the moment that the set of basis vectors  $\{\psi_i\}$  is fixed and known for a particular data vector  $\mathbf{x}$  then the LSAR interpolator can easily be extended to cover this case. The unknowns are now augmented by the basis coefficients,  $\{c_i\}$ . Define  $\mathbf{c}$  as a column vector containing the  $c_i$ 's and a  $(N \times Q)$  matrix  $\mathbf{G}$  such that  $\mathbf{x} = \mathbf{G}\mathbf{c} + \mathbf{r}$ , where  $\mathbf{r}$  is the vector of residual samples. The columns of  $\mathbf{G}$  are the basis vectors, i.e.  $\mathbf{G} = [\psi_1 \dots \psi_Q]$ . The excitation sequence can then be written in terms of  $\mathbf{x}$  and  $\mathbf{c}$  as  $\mathbf{e} = \mathbf{A}(\mathbf{x} - \mathbf{G}\mathbf{c})$ , which is the same form as for the general linear model (see section 4.1). As before the solution can easily be obtained from least squares, ML and MAP criteria, and the solutions will be equivalent in most cases. We consider here the least squares solution which minimises  $\mathbf{e}^T \mathbf{e}$  as before, but this time with respect to both  $\mathbf{x}_{(i)}$  and  $\mathbf{c}$ , leading to the following estimate:

$$\begin{bmatrix} \mathbf{x}_{(i)} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} & -\mathbf{A}_{(i)}^T \mathbf{A} \mathbf{G} \\ -\mathbf{G}^T \mathbf{A}^T \mathbf{A}_{(i)} & \mathbf{G}^T \mathbf{A}^T \mathbf{A} \mathbf{G} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} \\ \mathbf{G}^T \mathbf{A}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} \end{bmatrix} \quad (5.17)$$

This extended version of the interpolator reduces to the standard interpolator when the number of basis vectors,  $Q$ , is equal to zero. If we back-substitute for  $\mathbf{c}$  in (5.17), the following expression is obtained for  $\mathbf{x}_{(i)}$

alone:

$$\mathbf{x}_{(i)} = - \left( \mathbf{A}_{(i)}^T (\mathbf{I} - \mathbf{A}\mathbf{G}(\mathbf{G}^T \mathbf{A}^T \mathbf{A}\mathbf{G})^{-1} \mathbf{G}^T \mathbf{A}^T) \mathbf{A}_{(i)} \right)^{-1} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} - \mathbf{A}_{(i)} \mathbf{A}\mathbf{G}(\mathbf{G}^T \mathbf{A}^T \mathbf{A}\mathbf{G})^{-1} \mathbf{G}^T \mathbf{A}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} \right)$$

These two representations are equivalent to both the maximum likelihood (ML) and maximum *a posteriori* (MAP)<sup>1</sup> interpolator under the same conditions as the standard AR interpolator, i.e. that no missing samples occur in the first  $P$  samples of the data vector. In cases where missing data does occur in the first  $P$  samples, a similar adaptation to the algorithm can be made as for the pure AR case. The modified interpolator involves some extra computation in estimating the basis coefficients, but as for the pure AR case many of the terms can be efficiently calculated by utilising the banded structure of the matrix  $\mathbf{A}$ .

We do not address the issue of basis function selection here. Multiscale and ‘elementary waveform’ representations such as wavelet bases may capture the non-stationary nature of audio signals, while a sinusoidal basis is likely to capture the character of voiced speech and the steady-state section of musical notes. Some combination of the two may well provide a good match to general audio. Procedures have been devised for selection of the number and frequency of sinusoidal basis vectors in the speech and audio literature [127, 45, 66] which involve various peak tracking and selection strategies in the discrete Fourier domain. More sophisticated and certainly more computationally intensive methods might adopt a time domain model selection strategy for selection of appropriate basis functions from some large ‘pool’ of candidates. A Bayesian approach would be a strong possibility for this task, employing some of the powerful Monte Carlo variable selection methods which are now available [65, 108]. Similar issues of iterative AR parameter estimation apply as for the standard AR interpolator in the AR plus basis function interpolation scheme.

#### 5.2.3.2.1 Example: sinusoid+AR residual interpolation

As a simple example of how the inclusion of deterministic basis vectors can help in restoration performance we consider the interpolation of a short section of brass music, which has a strongly ‘voiced’ character, see figure 5.2. Figure 5.3 shows the same data with three missing sections, each of length 100 samples. This was used as the initialisation for the interpolation algorithm. Firstly a sinusoid + AR interpolation was applied, using 25 sinusoidal basis frequencies and an AR residual with order  $P = 15$ . The algorithm used was iterative, re-estimating the AR parameters, sinusoidal frequencies and missing data points at each step. The sinusoidal frequencies

---

<sup>1</sup>assuming a uniform prior distribution for the basis coefficients

are estimated rather crudely at each step by simply selecting the 25 frequencies in the DFT of the interpolated data which have largest magnitude. The number of iterations was 5. Figure 5.4 shows the resulting interpolated data, which can be seen to be a very effective reconstruction of the original uncorrupted data. Compare this with interpolation using an AR model of order 40 (chosen to match the 25+15 parameters of the sin+AR interpolation), as shown in figure 5.5, in which the data is under-predicted quite severely over the missing sections. Finally, a zoomed-in comparison of the two methods over a short section of the same data is given in figure 5.6, showing more clearly the way in which the AR interpolator under-performs compared with the sin+AR interpolator.

### 5.2.3.3 Random sampling methods

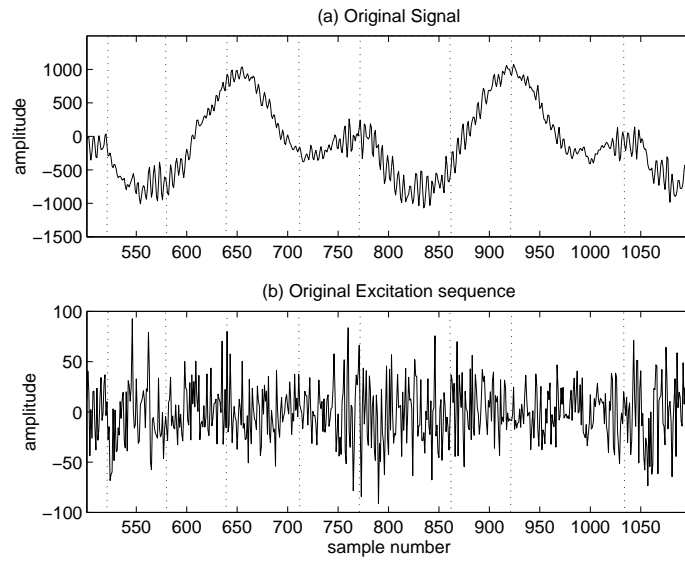
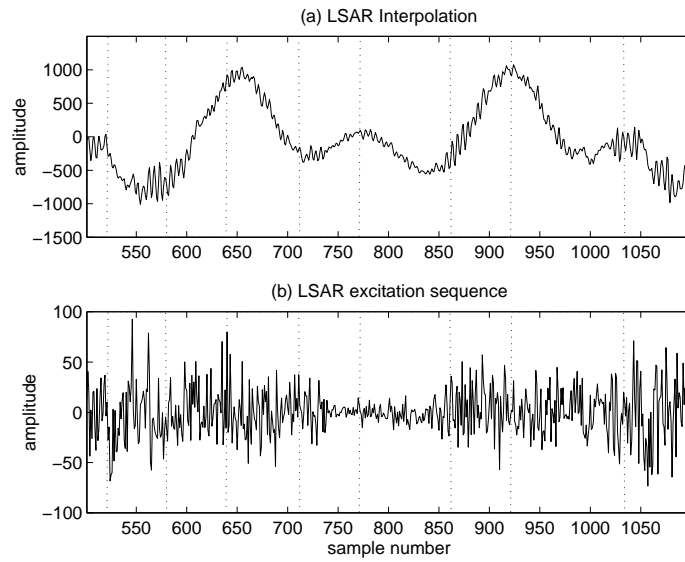
A further modification to the LSAR method is concerned with the characteristics of the excitation signal. We notice that the LSAR procedure seeks to minimise the excitation energy of the signal, irrespective of its time domain autocorrelation. This is quite correct, and desirable mathematical properties result. However, figure 5.8 shows that the resulting excitation signal corresponding to the corrupted region can be correlated and well below the level of surrounding excitation. As a result, the ‘most probable’ interpolants may under-predict the true signal levels and be over-smooth compared with the surrounding signal. In other words, ML/MAP procedures do not necessarily generate interpolants which are *typical* for the underlying model, which is an important factor in the *perceived* effect of the restoration. Rayner and Godsill [161] have devised a method which addresses this problem. Instead of minimising the excitation energy, we consider interpolants with constant excitation energy. The excitation energy may be expressed as:

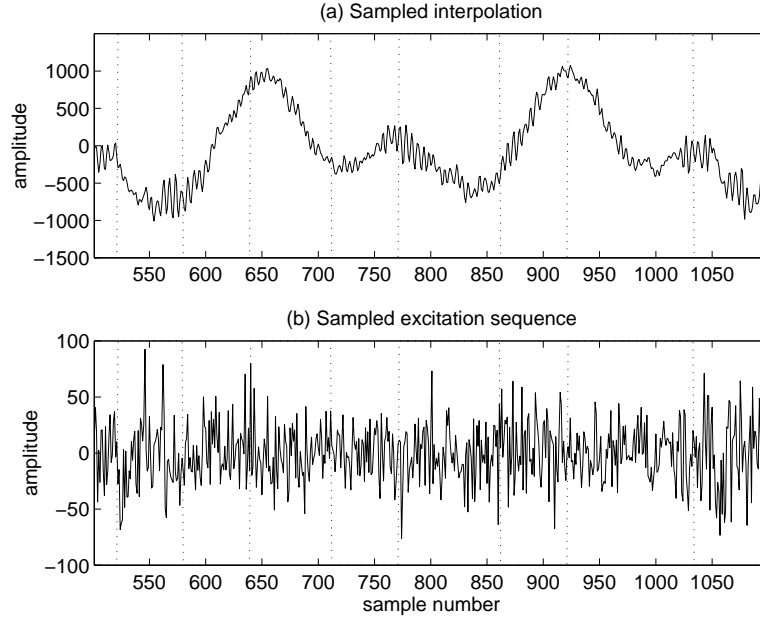
$$E = (\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^{\text{LS}})^T \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} (\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^{\text{LS}}) + E_{\text{LS}}, \quad E > E_{\text{LS}}, \quad (5.18)$$

where  $E_{\text{LS}}$  is the excitation energy corresponding to the LSAR estimate  $\mathbf{x}_{(i)}^{\text{LS}}$ . The positive definite matrix  $\mathbf{A}_{(i)}^T \mathbf{A}_{(i)}$  can be factorised into ‘square roots’ by Cholesky or any other suitable matrix decomposition [86] to give  $\mathbf{A}_{(i)}^T \mathbf{A}_{(i)} = \mathbf{M}^T \mathbf{M}$ , where  $\mathbf{M}$  is a non-singular square matrix. A transformation of variables  $\mathbf{u} = \mathbf{M}(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^{\text{LS}})$  then serves to de-correlate the missing data samples, simplifying equation (5.18) to:

$$E = \mathbf{u}^T \mathbf{u} + E_{\text{LS}}, \quad (5.19)$$

from which it can be seen that the (non-unique) solutions with constant excitation energy correspond to vectors  $\mathbf{u}$  with constant  $L_2$ -norm. The resulting interpolant can be obtained by the inverse transformation  $\mathbf{x}_{(i)} = \mathbf{M}^{-1} \mathbf{u} + \mathbf{x}_{(i)}^{\text{LS}}$ .

FIGURE 5.7. Original signal and excitation ( $P=100$ )FIGURE 5.8. LSAR interpolation and excitation ( $P = 100$ )

FIGURE 5.9. Sampled AR interpolation and excitation ( $P=100$ )

One suitable criterion for selecting  $\mathbf{u}$  might be to minimise the autocorrelation at non-zero lags of the resulting excitation signal, since the excitation is assumed to be white noise. This, however, requires a non-linear minimisation, and a practical alternative is to generate  $\mathbf{u}$  as Gaussian white noise with variance  $(E - E_{LS})/l$ , where  $l$  is the number of corrupted samples. The resulting excitation will have approximately the desired energy and uncorrelated character. A suitable value for  $E$  is the expected excitation energy for the AR model, provided this is greater than  $E_{LS}$ , i.e.  $E = \max(E_{LS}, N\sigma_e^2)$ .

Viewed within a probabilistic framework, the case when  $E = E_{LS} + l\sigma_e^2$ , where  $l$  is the number of unknown sample values, is equivalent to drawing a sample from the posterior density for the missing data,  $p(\mathbf{x}_{(i)} \mid \mathbf{x}_{-(i)}, \mathbf{a}, \sigma_e^2)$ . Figures 5.7-5.9 illustrate the principles involved in this sampled interpolation method. A short section taken from a modern solo vocal recording is shown in figure 5.7, alongside its estimated autoregressive excitation. The waveform has a fairly ‘noise-like’ character, and the corresponding excitation is noise-like as expected. The standard LSAR interpolation and corresponding excitation is shown in figure 5.8. The interpolated section (between the dotted vertical lines) is reasonable, but has lost the random noise-like quality of the original. Examination of the excitation signal shows that the LSAR interpolator has done ‘too good’ a job of minimising the excitation energy, producing an interpolant which, while optimal in a mean-



square error sense, cannot be regarded as typical of the autoregressive process. This might be heard as a momentary change in sound quality at the point of interpolation. The sampling-based interpolator is shown in figure 5.9. Its waveform retains the random quality of the original signal, and likewise the excitation signal in the gap matches the surrounding excitation. Hence the sub-optimal interpolant is likely to sound more convincing to the listener than the LSAR reconstruction.

Ó Ruanaidh and Fitzgerald [146, 166] have successfully extended the idea of sampled interpolates to a full Gibbs' Sampling framework [64, 63] in order to generate typical interpolates from the marginal posterior density  $p(\mathbf{x}_{(i)} \mid \mathbf{x}_{-(i)})$ . The method is iterative and involves sampling from the conditional posterior densities of  $\mathbf{x}_{(i)}$ ,  $\mathbf{a}$  and  $\sigma_e^2$  in turn, with the other unknowns fixed at their most recent sampled values. Once convergence has been achieved, the interpolation used is the last sampled estimate from  $p(\mathbf{x}_{(i)} \mid \mathbf{x}_{-(i)}, \mathbf{a}, \sigma_e^2)$ . See chapter 12 for a more detailed description and extensions to these methods.

#### 5.2.3.4 Incorporating a noise model

If we incorporate an i.i.d. Gaussian noise model as in (5.11) then the modified AR interpolator, assuming no corrupted samples within the first  $P$  elements of  $\mathbf{y}$ , is:

$$\mathbf{x}_{(i)}^{\text{MAP}} = - \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} + \frac{\sigma_e^2}{\sigma_v^2} \mathbf{I} \right)^{-1} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{x}_{-(i)} - \frac{\sigma_e^2}{\sigma_v^2} \mathbf{y}_{(i)} \right), \quad (5.20)$$

(see chapter 9), where  $\sigma_v^2$  is the variance of the corrupting noise, which is assumed independent and Gaussian. This interpolator has been found to give higher perceived restoration quality, particularly in the case of clicks with small amplitude. The assumed model and the resulting interpolator form a part of the Bayesian restoration schemes discussed in chapters 9-12. With this form of interpolator we now have the task of estimating the (unknown) noise variance  $\sigma_v^2$  in addition to the AR parameters and the excitation variance  $\sigma_e^2$ . This can be performed iteratively within the same type of framework as for the standard AR interpolator (section 5.2.2.4) and details of EM and MCMC based methods can be found in [83, 80, 82, 73, 72, 81, 129] and chapter 12.

#### 5.2.3.5 Sequential methods

The preceding algorithms all operate in a *batch* mode; that is, a whole vector of data is processed in one single operation. It will often be more convenient to operate in a *sequential* fashion, at each step inputting one new data point and updating our interpolation in the light of the information given by the new data point. The Kalman filter (see section 4.4) can be

used to achieve this. We first write the AR model in state space form:

$$\begin{aligned} y_n &= Z_n \alpha_n + u_n \\ \alpha_{n+1} &= T \alpha_n + H e_n \end{aligned}$$

where

$$\begin{aligned} \alpha_n &= [x_n \quad x_{n-1} \quad x_{n-2} \quad \dots \quad x_{n-P+1}]^T \\ T &= \begin{bmatrix} a_1 & a_2 & \dots & a_P \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix} \\ H &= [1 \quad 0 \quad \dots \quad 0]^T \end{aligned}$$

Here as before for the Kalman filter we have dropped our usual bold-face notation for matrices and vectors for the sake of simplicity, the dimensionality of each term being clear from its definition.

We allow the term  $Z_n$  to be time dependent in order to account for missing observations in a special way. For an uncorrupted data sequence with no missing data,  $Z_n = H^T$  for all  $n$  and the variance of  $u_n$  is zero, i.e. all the data points are exactly observed without error. For the case of a discarded or missing data point, we can set  $y_n = 0$  and  $Z_n = 0$  to indicate that the data point is not observed. The standard Kalman filter can then be run across the entire data sequence, estimating the states  $\alpha_n$  (and hence the interpolated data sequence  $x_n$ ) from the observed data  $y_0, \dots, y_n$ . The recursion is summarised as:

1. **Initialise:**  $a_0, P_0$ .
2. **Repeat for**  $n = 1$  **to**  $N$ :
  - (a) **Prediction:**

$$\begin{aligned} a_{n|n-1} &= T a_{n-1} \\ P_{n|n-1} &= T P_{n-1} T^T + H \sigma_e^2 H^T \end{aligned}$$

- (b) **Correction:**

$$\begin{aligned} a_n &= a_{n|n-1} + K_n (y_n - Z_n a_{n|n-1}) \\ P_n &= (I - K_n Z_n) P_{n|n-1} \end{aligned}$$

where

$$K_n = P_{n|n-1} Z_n^T (Z_n P_{n|n-1} Z_n^T + E[u_n^2])^{-1} \quad (\text{Kalman Gain})$$

The observation noise variance  $E[u_n^2]$  is zero for the missing data case considered so far, but we leave it in place for later inclusion of a Gaussian observation noise model. Assuming the AR model to be stable, the initialisation is given by  $a_0 = E[\alpha_0] = 0$  and  $P_0 = E[\alpha_0 \alpha_0^T]$ , where the latter term is the covariance matrix for  $P$  samples of the AR process (see appendix C). Note that many of the calculations in this recursion are redundant and can be eliminated by careful programming, since we know immediately that  $x_n = y_n$  for any uncorrupted samples with  $i_n = 0$ . This means that elements of  $P_m$  which involve such elements  $x_n$  can immediately be set to zero. Taking account of this observation the Kalman filter becomes a computationally attractive means of performing interpolation, particularly when missing samples do not occur in well spaced contiguous bursts so that the Toeplitz structure is lost in the batch based AR interpolator.

A straightforward sequential interpolator would extract at each time point the *last* element of  $a_n$ , which equals  $E[x_{n-P+1} | y_1, \dots, y_n]$ , since this is the element with the greatest degree of ‘lookahead’ or ‘smoothing’. This degree of lookahead may not be sufficient, especially when there are long bursts of corrupted samples in the data. To overcome the problem an augmented state space model can be constructed in which extra lagged data points are appended to the state  $\alpha_n$ . The Kalman filter equations retain the same form as above but interpolation performance is improved, owing to the extra lookahead in the estimated output.

We have shown how to implement a sequential version of the basic AR interpolator. By suitable modifications to the state transition and observation equations, which we do not detail here, it is possible also to implement using the Kalman filter all of the subsequent modifications to the AR interpolator discussed in later sections of the chapter, including the sampled versions and the AR plus basis function version.

#### 5.2.3.5.1 The Kalman Smoother

The Kalman filter may also be used to perform efficient batch-based interpolation. The filter is run forwards through the data exactly as above, but is followed by a backwards ‘smoothing’ pass through the data. The smoothing pass is a backwards recursion which calculates  $E[\alpha_n | y_1, \dots, y_N]$  for  $n = N - 1, N - 2, \dots, 1$ , i.e. the estimate of the state sequence conditional upon the *whole* observed data sequence. We do not give the details of the smoother here but the reader is referred for example to the texts by Anderson and Moore [5] and Harvey [91]. It should be noted that the results of such a procedure are then mathematically identical to the standard batch based AR interpolation.

### 5.2.3.5.2 Incorporating a noise model into the Kalman filter

We have assumed thus far that the observation noise is always zero, i.e.  $E[u_n^2] = 0$ . If we allow this term to be non-zero then a Gaussian noise model can be incorporated, exactly as for the batch based algorithms. Setting  $Z_n = H^T$  for all  $n$ ,  $E[u_n^2] = 0$  when  $i_t = 0$  and  $E[u_n^2] = \sigma_v^2$  when  $i_t = 1$  achieves the same noise model as in section 5.2.3.4. Interpolators based upon other noise models which allow some degree of noise at all sampling instants can be used to remove background noise and clicks jointly. These ideas have been applied in [141, 142] and will be returned to in the chapters concerning Bayesian restoration.

### 5.2.4 ARMA model-based interpolation

A natural extension to the autoregressive interpolation models is the autoregressive moving-average (ARMA) model (see section 4.2). In this model we extend the AR model by including a weighted sum of excitation samples as well as the usual weighted sum of output samples:

$$x_t = \sum_{i=1}^P a_i x_{n-i} + \sum_{j=0}^Q b_j e_{n-j}$$

with  $b_0 = 1$ . Here the  $\{b_i\}$  are the moving-average (MA) coefficients, which introduce zeros into the model as well as poles. This can be expected to give more accurate modelling of the spectral shape of general audio signals with fewer coefficients, although it should be noted that a very high order AR model can approximate any ARMA model arbitrarily well.

In this section we present an interpolator for ARMA signals which is based on work in a recent M.Eng. dissertation by Soon Leng Ng [140]. We adopt a MAP procedure, since the probability of the initial state vector is more crucial in the ARMA case than in the pure AR case. In principle we can express the ARMA interpolator for Gaussian signals within the earlier framework for Gaussian signals with known correlation structure, but there is some extra work involved in the calculation of the inverse covariance matrix for the ARMA case [21]. In order to obtain this covariance matrix we re-parameterise the ARMA process in terms of a ‘latent’ AR process  $\{u_n\}$  cascaded with a moving-average filter:

$$u_n = \sum_{i=1}^P a_i u_{n-i} + e_n \quad (5.21)$$

$$x_n = \sum_{j=0}^Q b_j u_{n-j} \quad (5.22)$$

A diagrammatic representation is given in figure 5.10.

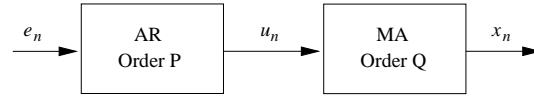


FIGURE 5.10. Equivalent version of ARMA model

The derivation proceeds by expressing the probability for  $\{x_t\}$  in terms of the probability of  $\{u_n\}$ , which we already have since it is an AR process. The linear transformation implied by (5.22) has unity Jacobian and hence the result is achieved by a straightforward change of variables. We then need only to account for the initial values of the latent process  $\mathbf{u}_0 = [u_{-P+1} \cdots u_{-1} u_0]^T$  which is carried out using the results of appendix C. Similar calculations can be found in time series texts, see e.g. [21, Appendix A7.3], although Box *et al.* derive their likelihoods in terms of initial states of both AR and MA components whereas we show that only the AR initial conditions are required. We then proceed to derive the MAP data interpolator in a similar fashion to the AR case.

For a data block with  $N$  samples equations (5.21) and (5.22) can be expressed in matrix form as follows:

$$\mathbf{e} = \mathbf{A}\mathbf{u} \quad (5.23)$$

$$\mathbf{x} = \mathbf{B}\mathbf{u} \quad (5.24)$$

where:

$$\mathbf{x} = [x_1 \cdots x_N]^T, \quad \mathbf{e} = [e_1 \cdots e_N]^T, \quad (5.25)$$

$$\mathbf{u} = [u_{-P+1} \cdots u_{-1} u_0 u_1 \cdots u_N]^T, \quad (5.26)$$

$\mathbf{A}$  is the  $N \times (N+P)$  matrix :

$$\mathbf{A} = \begin{bmatrix} -a_P & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\ 0 & -a_P & \cdots & -a_1 & 1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -a_P & \cdots & -a_1 & 1 \end{bmatrix} \quad (5.27)$$

and  $\mathbf{B}$  is the  $N \times (N+P)$  matrix:

$$\mathbf{B} = \begin{bmatrix} 0 & \cdots & 0 & b_Q & \cdots & b_1 & b_0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & b_Q & \cdots & b_1 & b_0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & b_Q & \cdots & b_1 & b_0 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 & b_Q & \cdots & b_1 & b_0 \end{bmatrix} \quad (5.28)$$

Suppose that  $\mathbf{u}$  is partitioned into  $\mathbf{u}_0$ , which contains the first  $P$  samples, and  $\mathbf{u}_1$  which contains the remaining  $N$  samples:

$$\mathbf{u}_0 = [u_{-P+1} \ u_{-P+2} \ \cdots \ u_0]^T \quad (5.29)$$

$$\mathbf{u}_1 = [u_1 \ u_2 \ \cdots \ u_N]^T \quad (5.30)$$

We can now partition the equations for  $\mathbf{e}$  and  $\mathbf{x}$  correspondingly:

$$\mathbf{e} = \mathbf{A}_0 \mathbf{u}_0 + \mathbf{A}_1 \mathbf{u}_1 \quad (5.31)$$

$$\mathbf{x} = \mathbf{B}_0 \mathbf{u}_0 + \mathbf{B}_1 \mathbf{u}_1 \quad (5.32)$$

where  $\mathbf{A} = [\mathbf{A}_0 \ \mathbf{A}_1]$  and  $\mathbf{B} = [\mathbf{B}_0 \ \mathbf{B}_1]$  are columnwise partitions of  $\mathbf{A}$  and  $\mathbf{B}$ .

$\mathbf{B}_1$  is invertible since it is a lower triangular square matrix and  $b_0 = 1$ . Thus we can rewrite equation (5.32) as:

$$\mathbf{u}_1 = \mathbf{B}_1^{-1}(\mathbf{x} - \mathbf{B}_0 \mathbf{u}_0) \quad (5.33)$$

and back-substitute into (5.31) to give:

$$\mathbf{e} = \mathbf{C} \mathbf{u}_0 + \mathbf{D} \mathbf{x} \quad (5.34)$$

where

$$\mathbf{C} = \mathbf{A}_0 - \mathbf{A}_1 \mathbf{B}_1^{-1} \mathbf{B}_0 \quad \text{and} \quad \mathbf{D} = \mathbf{A}_1 \mathbf{B}_1^{-1} \quad (5.35)$$

The probability of  $\mathbf{u}_1$  given  $\mathbf{u}_0$ , the conditional probability for the latent AR process (see appendix C), is:

$$p(\mathbf{u}_1 | \mathbf{u}_0) \propto \exp \left( -\frac{1}{2\sigma_e^2} \mathbf{e}^T \mathbf{e} \right) \Big|_{\mathbf{e}=\mathbf{A}\mathbf{u}}$$

(the conditioning upon ARMA parameters  $\mathbf{a}$  and  $\mathbf{b}$  is implicit from here on). Since  $\mathbf{B}_1$  is lower triangular with unity diagonal elements ( $b_0 = 1$ ) the Jacobian of the transformation from  $\mathbf{u}_1$  to  $\mathbf{x}$  conditioned upon  $\mathbf{u}_0$ , as defined by (5.33), is unity and we can write immediately:

$$p(\mathbf{x} | \mathbf{u}_0) \propto \exp \left( -\frac{1}{2\sigma_e^2} \mathbf{e}^T \mathbf{e} \right) \Big|_{\mathbf{e}=\mathbf{C}\mathbf{u}_0+\mathbf{D}\mathbf{x}}$$

We have also the probability of  $\mathbf{u}_0$ , again from appendix C:

$$p(\mathbf{u}_0) \propto \exp \left( -\frac{1}{2\sigma_e^2} \mathbf{u}_0^T \mathbf{M}_0^{-1} \mathbf{u}_0 \right)$$

where  $\mathbf{M}_0$  is the covariance matrix for  $P$  samples of the AR process with unity excitation variance. Hence the joint probability is:

$$\begin{aligned} p(\mathbf{x}, \mathbf{u}_0) &= p(\mathbf{x} | \mathbf{u}_0) p(\mathbf{u}_0) \\ &\propto \exp \left( -\frac{1}{2\sigma_e^2} ((\mathbf{C}\mathbf{u}_0 + \mathbf{D}\mathbf{x})^T (\mathbf{C}\mathbf{u}_0 + \mathbf{D}\mathbf{x}) + \mathbf{u}_0^T \mathbf{M}_0^{-1} \mathbf{u}_0) \right). \end{aligned}$$

$p(\mathbf{x})$  is now obtained by integration over the unwanted component  $\mathbf{u}_0$ :

$$\begin{aligned} p(\mathbf{x}) &= \int_{\mathbf{u}_0} p(\mathbf{x}, \mathbf{u}_0) d\mathbf{u}_0 \\ &\propto \exp \left( -\frac{1}{2\sigma_e^2} (\mathbf{x}^T \mathbf{D}^T (\mathbf{I} - \mathbf{C}(\mathbf{C}^T \mathbf{C} + \mathbf{M}_0^{-1})^{-1} \mathbf{C}^T) \mathbf{D} \mathbf{x}) \right) \end{aligned}$$

where the integral is performed using the multivariate Gaussian integral result (see appendix A.5). The resulting distribution is multivariate Gaussian, as expected, and by comparison with the general case (A.2) we obtain the inverse covariance matrix as:

$$\sigma_e^2 \mathbf{R}_x^{-1} = \mathbf{D}^T (\mathbf{I} - \mathbf{C}(\mathbf{C}^T \mathbf{C} + \mathbf{M}_0^{-1})^{-1} \mathbf{C}^T) \mathbf{D}$$

We can now substitute this expression for the inverse covariance matrix into the general expression of equation (5.8) to find the MAP ARMA interpolator. The interpolation involves significantly more calculations than the AR-based interpolator (equation 5.15) although it should be noted that calculation of  $\mathbf{B}_1^{-1}$ , which is required in (5.35), takes only  $NQ/2$  flops since  $\mathbf{B}_1$  is Toeplitz, banded and lower triangular. Furthermore, many of the matrix multiplication operations such as  $\mathbf{A}_1$  can be realised using simple filtering operations. There is always, however, the overhead of inverting the  $(P \times P)$  matrix  $(\mathbf{C}^T \mathbf{C} + \mathbf{M}_0^{-1})$ .

As an alternative to the batch based method outlined above, the ARMA interpolator can be realised sequentially using the Kalman filter in a very similar way to the AR case (see section 5.2.3.5). The ARMA model can be expressed in state space form (see e.g. [5, 91]) and having done so the Kalman filter recursions follow exactly as before on the modified state space model.

#### 5.2.4.1 Results from the ARMA interpolator

##### 5.2.4.1.1 Synthetic ARMA process

The performance of the ARMA model-based interpolator is evaluated here using two audio signals, each of length roughly 10 seconds: a piece of fast moving piano music ('fast') and a section of solo vocal singing ('diner'). Both signals were sampled at 44.1 kHz and sections of data were removed at periodic intervals from the original signal. Each missing section was 50 samples long and approximately 10% of the data was removed. The AR interpolator assumed an AR order of 15 while the ARMA interpolator assumed an ARMA order of (15, 5). The model parameters in both cases were estimated using MATLAB's in-built ARMA function 'armax' operating on the uncorrupted input data, allowing the parameters to change once every 1000 data points. The mean squared error (MSE) and maximum absolute deviation (MAD) in each section were calculated and the average values are tabulated in table 5.1.

Signal		Average MSE	Average MAD
'fast'	AR	$6.6428 \times 10^9$	923.74
	ARMA	$6.4984 \times 10^5$	911.26
'diner'	AR	$3.6258 \times 10^9$	844.52
	ARMA	$3.0257 \times 10^5$	790.59

TABLE 5.1. Comparison of error performance for tracks 'fast' and 'diner'.

The errors show that the ARMA interpolator performs better than the LSAR interpolator for both types of audio signals under these conditions, although the difference is by no means dramatic. The signals restored by the ARMA interpolator were also subjectively evaluated by informal listening tests and in both cases the differences from the original (clean) audio signals were almost inaudible, but the differences between the two interpolation methods were also extremely subtle. These results indicate that it may not be worthwhile using extra computation to interpolate using ARMA models.

### 5.2.5 Other methods

Several transform-domain methods have been developed for click replacement. Montresor, Valière and Baudry [135] describe a simple method for interpolating wavelet coefficients of corrupted audio signals, which involves substituting uncorrupted wavelet coefficients from nearby signal according to autocorrelation properties. This, however, does not ensure continuity of the restored waveform and is not a localised operation in the signal domain. An alternative method, based in the discrete Fourier domain, which is aimed at restoring long sections of missing data is presented by Maher [118]. In a similar manner to the sinusoidal modelling methods of McAulay and Quatieri [126, 158, 127], this technique assumes that the signal is composed as a sum of sinusoids with slowly varying frequencies and amplitudes. Spectral peak 'tracks' from either side of the gap are identified from the Discrete Fourier Transform (DFT) of successive data blocks and interpolated in frequency and amplitude to generate estimated spectra for the intervening material. The inverse DFTs of the missing data blocks are then inserted back into the signal. The method is reported to be successful for gap lengths of up to 30ms, or well over 1000 samples at audio sampling rates. A method for interpolation of signals represented by the multiple sinusoid model is given in [192, Chapter 6].

Godsill and Rayner [77, 70] have derived an interpolation method which operates in the frequency domain. This can be viewed as an alternative to the LSAR interpolator (see section 5.2.2) in which power spectral density (PSD) information is directly incorporated in the frequency domain. Real and imaginary DFT components are modelled as independent Gaussians with variance proportional to the PSD at each frequency. These assump-



tions of independence are known to hold exactly for periodic random processes [173], so the method is best suited to musical signals with strongly tonal content which are close to periodic. The method can, however, also be used for other stationary signals provided that a sufficiently long block length is used (e.g. 500-2000 samples) since the assumptions also improve as block length increases [148]. The Maximum *a posteriori* solution is of a similar form and complexity to the LSAR interpolator, and is potentially useful as an alternative to the other methods when the signal has a quasi-periodic or tonal character. A robust estimate is required for the PSD, and this can usually be obtained through averaged DFTs of the surrounding clean data, although iterative methods are also possible, as in the case of the LSAR estimator.

### 5.3 Detection of clicks

In the last section we discussed methods for interpolation of corrupted samples. All of these methods assumed complete knowledge of the position of click-type corruption in the audio waveform. In practice of course this information is completely unknown *a priori* and some kind of detection procedure must be applied in order to extract the timing of the degradation. There are any number of ways by which click detection can be performed, ranging from entirely *ad hoc* filtering methods through to model based probabilistic detection algorithms, as described in later chapters. In this section we describe some simple techniques which have proved extremely effective in most gramophone environments. The discussion cannot ever be complete, however, as these algorithms are under constant development and it is always possible to devise some clever trick which will improve slightly over the current state of the art!

Click detection for audio signals involves the identification of samples which are not drawn from the underlying clean audio signal; in other words they are drawn from some spurious ‘outlier’ distribution. There are close relationships between click detection and work in robust parameter estimation and treatment of outliers in data analysis, from fields as diverse as medical signal processing, underwater signal processing and statistical data analysis. In the statistical field in particular there has been a vast amount of work in the treatment of outliers (see e.g. [13, 12] for extensive review material, and further references in the later chapters on statistical click removal techniques). Various criteria for detection are possible, including minimum probability of error and related concepts, but strictly speaking the aim of any audio restoration scheme is to remove only those artefacts which are audible to the listener. Any further processing is not only unnecessary but will increase the chance of distorting the perceived signal quality. Hence a truly optimal system should take into account the trade-

off between the audibility of artefacts and perceived distortion as a result of processing, and will involve consideration of complex psychoacoustical effects in the human ear (see e.g. [137]). Such an approach, however, is difficult both to formulate and to realise, so we limit discussion here only to criteria which are currently well understood in a mathematical sense. This is not to say that new algorithms should not strive to attain the higher target of a *perceptually* optimal restoration.

The simplest click detection methods involve a high-pass filtering operation on the signal, the assumption being that most audio signals contain little information at high frequencies, while clicks, like impulses, have spectral content at all frequencies. Clicks are thus enhanced relative to the signal by the high-pass filtering operation and can easily be detected by thresholding the filtered output. The method has the advantage of being simple to implement and having no unknown system parameters (except for a detection threshold). This principle is the basis of most analogue de-clicking equipment [34, 102] and some simple digital click detectors [101]. Of course, the method will fail if the audio signal itself has strong high frequency content or the clicks are band-limited. Along similar lines, wavelets and multiresolution methods in general [3, 37, 38] have useful localisation properties for singularities in signals (see e.g. [120]), and a Wavelet filter at a fine resolution can be used for the detection of clicks. Such methods have been studied and demonstrated successfully by Montresor, Valière *et al.* [185, 135].

Other methods attempt to incorporate prior information about signal and noise into a model-based detection procedure. We will focus upon such methods since we believe that the best results can be achieved through the use of all the prior information which is available about the problem. We will describe techniques based upon AR modelling assumptions for the audio signal, the principles of which can easily be extended to some of the other models already encountered.

### 5.3.1 Autoregressive (AR) model-based click detection

Techniques for detection and removal of impulses from autoregressive (AR) signals have been developed in other fields of signal processing from robust filtering principles (see e.g. [9, 53]). These methods apply non-linear functions to the autoregressive excitation sequence in order to make parameter estimates robust in the presence of impulses and allow detection of impulses.

Autoregressive detection methods in the audio restoration field originated with Vaseghi and Rayner [190, 187, 191]. A sub-frame of the underlying audio data  $\{x_t; t = 1, \dots, N\}$  is assumed to be drawn from a short-term stationary autoregressive (AR) process, as for the AR-based interpolators

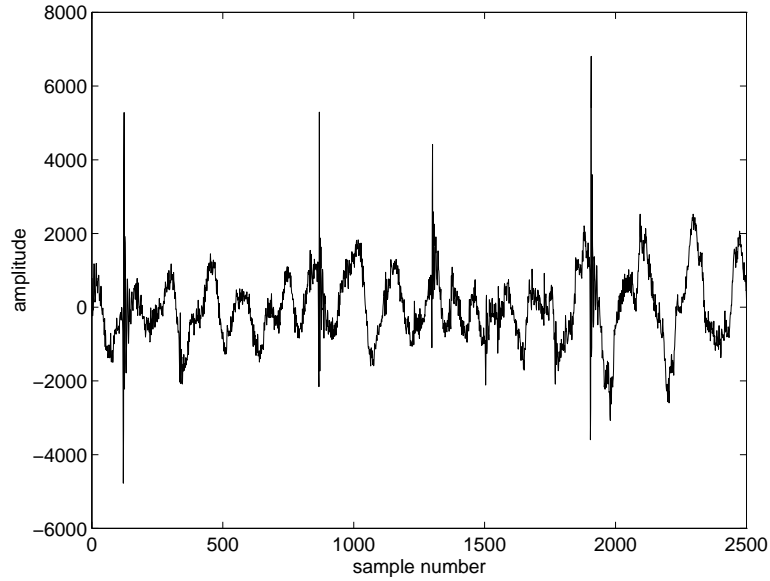
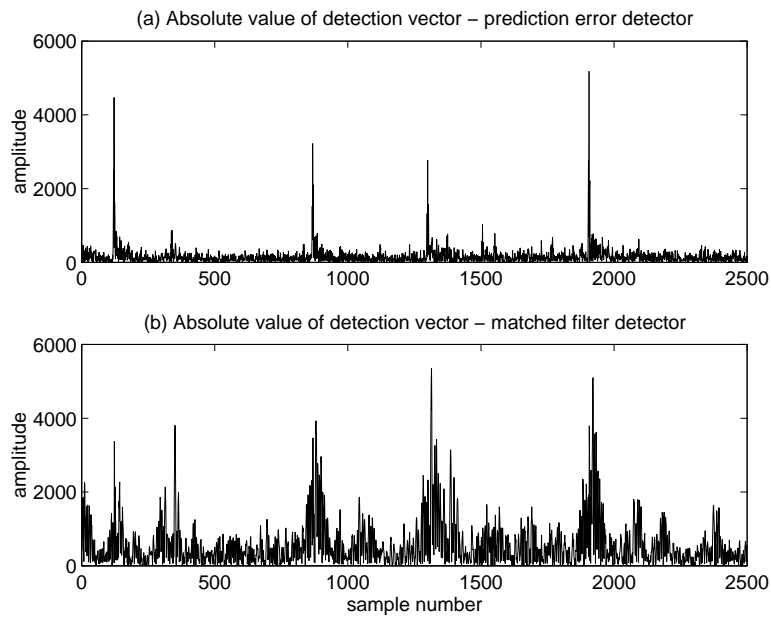


FIGURE 5.11. Click-degraded Music Waveform taken from 78rpm recording

FIGURE 5.12. AR-based detection for above waveform,  $P=50$ . (a) Prediction error filter (b) Matched filter.

of the previous section:

$$x_t = \sum_{i=1}^P a_i x_{n-i} + e_t$$

The prediction error  $e_t$  should take on small values for most of the time, especially if Gaussian assumptions can be made for  $\{e_t\}$ . Now, consider what happens when an impulse replaces the signal. Of course the impulse will be unrelated to the surrounding signal values and it is likely to cause a large error if an attempt is made to predict its value from the previous values of  $x_t$ . Hence, if we apply the inverse AR filter to an impulse-corrupted AR signal  $y_t$  and observe the output prediction error sequence  $\epsilon_t = y_t - \sum_{i=1}^P a_i y_{n-i}$  we can expect large values at times when impulses are present and small values for the remainder of the time. This is the basic principle behind the simple AR model-based detectors.

If we suppose that the AR parameters and corresponding prediction error variance  $\sigma_e^2$  are known, then a possible detection scheme would be:

For  $n = 1$  to  $N$

1. Calculate prediction error:  $\epsilon_t = y_t - \sum_{i=1}^P a_i y_{n-i}$
2. if  $|\epsilon_t| > k\sigma_e$  then  $i_t = 1$ , else  $i_t = 0$

end

Here  $k$  is a detection threshold which might be set to around 3 if we believed the excitation  $\{e_t\}$  to be Gaussian and applied the standard ‘ $3\sigma$ ’ rule for detection of unusual events.  $i_t$  is a binary detection indicator as before.

In practice the AR model parameters  $\mathbf{a}$  and the excitation variance  $\sigma_e^2$  are unknown and must be estimated from the corrupted data  $y_t$  using some procedure robust to impulsive noise. Robust methods for this estimation procedure are well-known (see e.g. [95, 139, 123]) and can be applied here, at least to give initial estimates of the unknown parameters. The detection process can then be iterated a few times if time permits, each time performing interpolation of the detected samples and re-estimation of the AR parameters from that restored output. The whole process can be repeated frame by frame through an entire track of audio material, leading to a click-reduced output.

### 5.3.1.1 Analysis and limitations

Some basic analysis of the AR detection method is instructive. Substituting for  $y_t$  from (5.1) using (4.41) gives:

$$\epsilon_t = e_t + i_t n_t - \sum_{i=1}^P i_{t-i} n_{t-i} a_i \quad (5.36)$$

which is composed of the true signal excitation  $e_t$  and a weighted sum of present and past impulsive noise values. If  $\{x_t\}$  is zero mean and has variance  $\sigma_x^2$  then  $e_t$  is white noise with variance  $\sigma_e^2 = 2\pi \frac{\sigma_x^2}{\int_{-\pi}^{\pi} \frac{1}{|A(e^{j\theta})|^2} d\theta}$ , where

$A(z) = 1 - \sum_{i=1}^P a_i z^{-i}$ . The reduction in power here from signal to excitation can be 40dB or more for highly correlated audio signals. Consideration of (5.36), however, shows that a single impulse contributes the impulse response of the prediction error filter, weighted by the impulse amplitude, to the detection signal  $\epsilon_t$ , with maximum amplitude corresponding to the maximum in the impulse response. This means that considerable amplification of the impulse relative to the signal can be achieved for all but uncorrelated, noise-like signals. It should be noted, however, that this amplification is achieved at the expense of localisation in time of the impulse, whose effect is now spread over  $P + 1$  samples of the detection signal  $\epsilon_t$ . This will have adverse consequences when a number of impulses is present in the same vicinity, since their impulse responses may cancel one another out or add constructively to give false detections. More generally, threshold selection will be troublesome when impulses of widely differing amplitudes are present, since a low threshold which is appropriate for very small clicks will lead to false detections in the  $P$  detection values which follow a large impulse.

Detection can then be performed by thresholding  $|\epsilon_t|$  to identify likely impulses. Choice of threshold will depend upon the AR model, the variance of  $\{e_t\}$  and the size of impulses present, and will reflect trade-offs between false and missed detection rates. Optimal thresholds can be obtained under certain assumptions for the noise and signal statistics, see [69, appendix H], but will in more general conditions be a difficult issue. See figure 5.12(a) for a typical example of the detection prediction error  $\epsilon_t$  obtained using this method, which shows how the impulsive interference is strongly amplified relative to the signal component.

### 5.3.1.2 Adaptations to the basic AR detector

As the previous discussion will have indicated, the most basic form of AR detector described thus far is far from ideal. Depending upon the threshold chosen it will either leave a residual click noise behind or it will cause perceptual damage to the underlying audio signal. A practical scheme will

have to devise improvements aimed at overcoming the limitations discussed above. We do not detail the precise structure of such an algorithm as most of the ideas are *ad hoc*, but rather point out some areas where improvements can be achieved:

- **Clicks occur as ‘bursts’ of corruption.** It is very rare for a single impulse to occur in the corrupted waveform. It is much more common for clicks to have some finite width, say between 5 and 100 samples at 44.1kHz sampling rates. This would correspond to the width of the physical scratch or irregularity in the recorded medium. Hence improvements can be achieved by allowing a ‘buffer’ zone of a few samples both before and after the clicks detected as above. There are many ways for achieving this, including pre- and post-filtering of the prediction error or the detection vector with a non-linear ‘pulse broadening’ filter.
- **Backward prediction error.** Very often a good estimate can be obtained for the position of the start of a click, but it is more difficult to distinguish the end of a click, owing in part to the forward ‘smearing’ effect of the prediction error filter (see figure 5.12(a)). This can be alleviated somewhat by using information from the backward prediction error  $\epsilon_t^b$ :

$$\epsilon_t^b = y_t - \sum_{i=1}^P a_i y_{n+i}$$

$\epsilon_t^b$  can often give clean estimates of the end point of a click, so some appropriate combination of  $\epsilon_t$  and  $\epsilon_t^b$  will improve detection capability.

### 5.3.1.3 Matched filter detector

An adaptation of the basic AR detection method, also devised by Vaseghi and Rayner, considers the impulse detection problem from a matched filtering perspective [186]. The ‘signal’ is the impulse itself, while the autoregressive audio data is regarded as coloured additive noise. The prediction error filter described above can then be viewed as a pre-whitening stage for the autoregressive noise, and the full matched filter is given by  $A(z)A(z^{-1})$ , a non-causal filter with  $2P + 1$  coefficients which can be realised with  $P$  samples of lookahead. The matched filtering approach provides additional amplification of impulses relative to the signal, but further reduces localisation of impulses for a given model order. Choice between the two methods will thus depend on the range of click amplitudes present in a particular recording and the degree of separation of individual impulses in the waveform. See figure 5.12(b) for an example of detection using the matched

filter. Notice that the matched filter has highlighted a few additional impulse positions, but at the expense of a much more ‘smeared’ response which will make accurate localisation of the clicks more difficult. Hence the prediction-error detector is usually preferred in practice.

Both the prediction error detection algorithm and the matched filtering algorithm are efficient to implement and can be operated in real time using DSP microprocessors. Results of a very high standard can be achieved if a careful strategy is adopted for extracting the precise click locations from the detection signal.

#### 5.3.1.4 Other models

The principles of the AR-based detector can be extended to some of the other audio models which were used for interpolation. For example, an ARMA model can easily replace the AR model in the detection step by applying the inverse ARMA model to the corrupted data in place of the AR prediction error filter. We have found that this can give improved localisation of clicks, although it must of course be ensured that the ARMA model is itself invertible. The advantages of the sin+AR residual model can also be exploited by performing AR-based detection directly on the estimated AR residual. This can give a greater sensitivity to small clicks and crackles than is obtainable using the standard AR-based detector and also prevents some signal-damaging false alarms in the detection process.

## 5.4 Statistical methods for the treatment of clicks

The detection and replacement techniques described in the preceding sections can be combined to give very successful click concealment, as demonstrated by a number of research and commercial systems which are now used for the re-mastering of old recordings. However, some of the difficulties outlined above concerning the ‘masking’ of smaller defects by large defects in the detection process, the poor time localisation of some detectors in the presence of impulse ‘bursts’ and the inadequate performance of existing interpolation methods for certain signal categories, has led to further research which considers the problem from a more fundamental statistical perspective [76, 79, 83, 82]. These methods model explicitly both signal and noise sources, using rigorous statistical methods to perform more accurate and robust removal of clicks. The basis of these techniques can be found in chapters 9 - 12. The same framework may be extended to perform joint removal of clicks and background noise in one single procedure, and some recent work on this problem can be found in [81] for autoregressive signals and in [72, 73] for autoregressive moving-average (ARMA) signals.

A fully model based statistical methodology for treatment of audio defects (not just clicks and crackles) has many advantages. One significant

drawback, however, is the increase in computational requirements which results from the more sophisticated treatment. This becomes less of a problem as the speed and memory size of computer systems improves.

## 5.5 Discussion

In this chapter we have seen many possible methods for interpolation and detection of clicks in audio signals. The question remains as to which can be recommended in the practical situation. In answer to this it should be noted that all of the methods given have their advantages for particular types of signal, but that not all will perform well with audio signals in their generality; this is because some of the models used are aimed at signals of a particular type, such as periodic or pitch-based. The most generally applicable of the methods are the autoregressive (AR) based techniques, which can be applied quite successfully to most types of audio. The basic AR scheme can, however, lead to audible distortion of strongly ‘voiced’ musical extracts such as brass and vocal music. We now briefly summarise the benefits and drawbacks of the other techniques described in the chapter.

The pitch based adaptation of the basic AR scheme can give considerably better reconstructions of pitched voice and instrumental waveforms, but requires a robust estimate of the pitch period and may not be successful in unvoiced sections. The AR + basis function interpolators are more generally applicable and a simple choice of sinusoidal basis leads to successful results and improved detection compared with a pure AR model. Random sampling methods as outlined here lead to a modest improvement in interpolation of long gap lengths. They are more important, however, as the basis of some of the more sophisticated statistical methods of later chapters. The same comments apply to the interpolation methods which incorporate a noise model. Sequential methods using the Kalman filter can be applied to all of the models considered in the chapter and are an implementation detail which may be important in certain real time applications. The ARMA model has been found to give some improvements in detection and interpolation, but the small improvements probably do not justify the extra computational load.

Statistical methods were briefly mentioned in the chapter and will be considered in more detail in chapters 9 - 12. These methods employ explicit models of both the signal and degradation process, which allows a rigorous model-based methodology to be applied to both detection and interpolation of clicks. We believe that methods such as these will be a major source of future advances in the area of click treatment.



# 6

## Hiss Reduction

Random additive background noise is a form of degradation common to all analogue measurement, storage and recording systems. In the case of audio signals the noise, which is generally perceived as ‘hiss’ by the listener, will be composed of electrical circuit noise, irregularities in the storage medium and ambient noise from the recording environment. The combined effect of these sources will generally be treated as one single noise process, although we note that a pure restoration should strictly not treat the ambient noise, which might be considered as a part of the original ‘performance’. Random noise generally has significant components at all audio frequencies, and thus simple filtering and equalisation procedures are inadequate for restoration purposes.

Analogue tape recordings typically exhibit noise characteristics which are stationary and for most purposes white. At the other end of the scale, many early 78rpm and cylinder recordings exhibit highly non-stationary coloured noise characteristics, such that the noise can vary considerably within each revolution of the playback system. This results in the characteristic ‘swishing’ effect associated with some early recordings. In recording media which are also affected by local disturbances, such as clicks and low frequency noise resonances, standard practice is to restore these defects prior to any background noise treatment.

Noise reduction has been of great importance for many years in engineering disciplines. The classic least-squares work of Norbert Wiener [199] placed noise reduction on a firm analytic footing, and still forms the basis of many noise reduction methods. In the field of speech processing a large number of techniques has been developed for noise reduction, and many of

these are more generally applicable to noisy audio signals. We do not attempt here to describe every possible method in detail, since these are well covered in speech processing texts (see for example [112, 110] and numerous subsequent texts). We do, however, consider some standard approaches which are appropriate for general audio signals and emerging techniques which are likely to be of use in future work. It is worth mentioning that where methods are derived from speech processing techniques, as in for example the spectral attenuation methods of section 6.1, sophisticated modifications are required in order to match the stringent fidelity requirements and signal characteristics of an audio restoration system.

## 6.1 Spectral domain methods

Certainly the most popular methods for noise reduction in audio signals to date are based upon short-time processing in the spectral domain. The reason for the success of these methods is that audio signals are usually composed of a number of line spectral components which correspond to fundamental pitch and partials of the notes being played. Although these line components are time-varying they can be considered as fixed over a short analysis window with a duration of perhaps 0.02 seconds or more. Hence analysing short blocks of data in the frequency domain will concentrate the signal energy into a relatively few frequency ‘bins’ with high signal-to-noise ratio. This means that noise reduction can be performed in the frequency domain while maintaining the important parts of the signal spectrum largely unaffected. Processing is performed in a transform domain, usually the discrete Fourier Transform (DFT)  $Y(n, m)$  of sub-frames in the observed data  $\{y_n\}$ :

$$Y(n, m) = \sum_{l=0}^{N-1} g_l y_{(nM+l)} \exp(-jlm2\pi/N), \quad m = 0, \dots, N-1.$$

Index  $n$  is the sub-frame number within the data and  $m$  is the frequency component number.  $g_l$  is a time domain pre-windowing function with suitable frequency domain characteristics, such as the Hanning or Hamming Window.  $N$  is the sub-frame length and  $M < N$  is the number of samples between successive sub-frames. Typically  $M = N/2$  or  $M = N/4$  to allow a significant overlap between successive frames. A suitable value of  $N$  which encompasses a large enough window of data while not seriously violating the assumption of short-term fixed frequency components will be chosen, and  $N = 1024$  or  $N = 2048$  are appropriate choices which allow calculation with the FFT. Cappé [30, 32] has performed analysis which justifies the use of block lengths of this order.

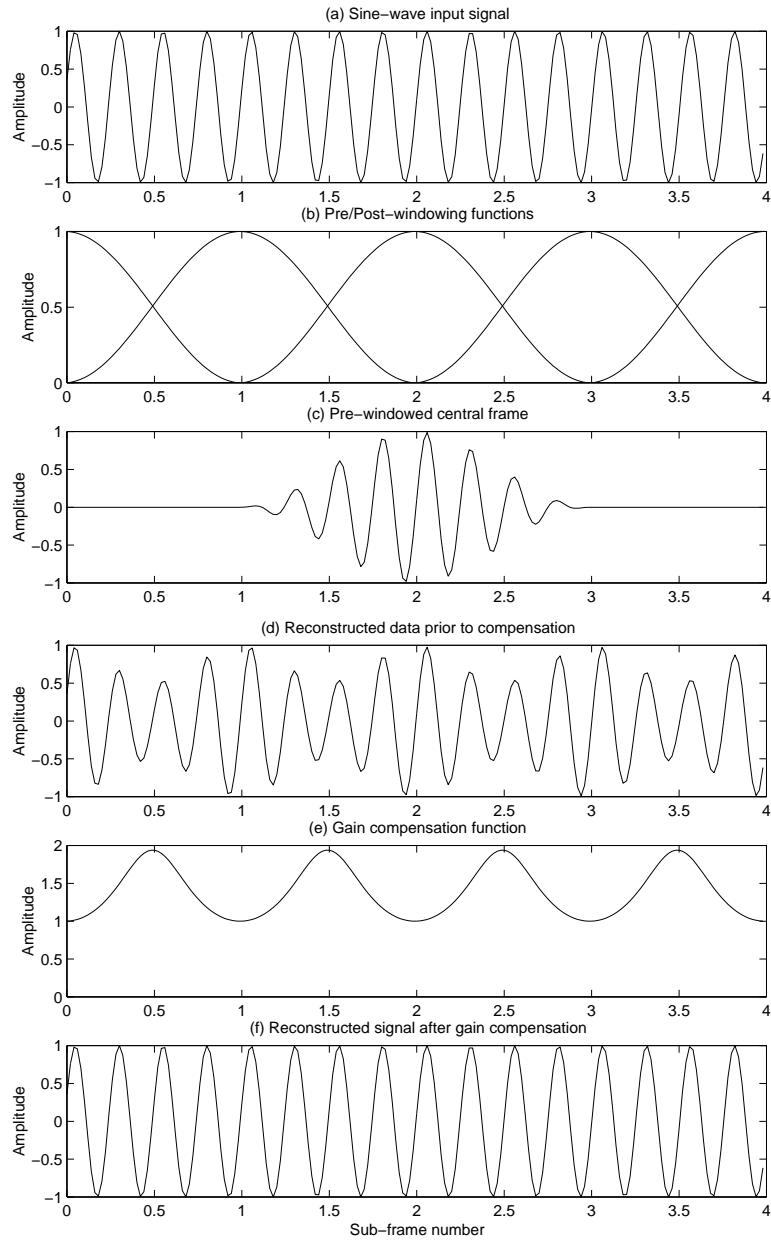


FIGURE 6.1. a) Input sine wave  $x_t$  b) Pre-/Post-windowing functions  $g_t$  and  $h_t$  (c) Pre-windowed data (d) Reconstruction without gain compensation (e) Gain compensation window  $w_t$  (f) Reconstructed sine wave

Processing is then performed on the spectral components  $Y(n, m)$  in order to estimate the spectrum of the ‘clean’ data  $X(n, m)$ :

$$\hat{X}(n, m) = f(Y(n, m))$$

where  $f(\cdot)$  is a function which performs noise reduction on the spectral components. We will discuss some possible forms for  $f(\cdot)$  in the next section.

The estimated spectrum  $\hat{X}(n, m)$  is then inverse DFT-ed to obtain a time domain signal estimate for sub-frame  $n$  at sample number  $nM + l$ :

$$\hat{x}_{nM+l}^n = h_l \frac{1}{N} \sum_{m=0}^{N-1} \hat{X}(n, m) \exp(jlm2\pi/N), \quad l = 0, \dots, N-1$$

where  $h_l$  is a post-processing window function which is typically tapered to zero at its ends to ensure continuity of restored output. Again, Hamming or Hanning windows would be suitable choices here.

Finally, the reconstructed signal in sub-frame  $n$  is obtained by an overlap-add method:

$$\hat{x}_{nM+l} = w_l \sum_{m \in \mathcal{M}} \hat{x}_{nM+l}^m$$

where  $\mathcal{M} = \{m; mM \leq nM + l < mM + N\}$ ,  $w_l$  is a weighting function which compensates for the overall gain of the pre- and post-windowing functions  $g_l$  and  $h_l$ :

$$w_l = \frac{1}{\sum_{m \in \mathcal{M}_0} g_{mM+l} h_{mM+l}}$$

and  $\mathcal{M}_0 = \{m; 0 \leq mM + l < N\}$ . The summation sets in these two expressions are over all adjacent windows which overlap with the current output sample number.

This time domain pre-windowing, post-windowing and gain compensation procedure is illustrated for Hanning windows with 50% block overlap ( $M = N/2$ ) in figure 6.1. The processing function is switched off, i.e.  $f(Y) = Y$ , so we wish in this case to reconstruct the input signal unmodified at the output.

### 6.1.1 Noise reduction functions

The previous section has described a basic scheme for analysis, processing and resynthesis of noisy audio signals using the DFT, a similar method to that first presented by Allen [4]. We have not yet, however, detailed the type of processing functions  $f(\cdot)$  which might be used to perform noise reduction for the spectral components. Many possible variants have been proposed in the literature, some based on heuristic ideas and others on a more rigorous basis such as the Wiener, maximum likelihood or maximum *a posteriori* estimation. See [20] for a review of the various suppression rules from a statistical perspective.

## 6.1.1.1 The Wiener solution

If the noise is assumed to be additive and independent of the signal, then the frequency domain Wiener filter which minimises the mean-squared error of the time domain reconstruction is given by [148]:

$$H(\omega) = \frac{\mathcal{S}_X(\omega)}{\mathcal{S}_X(\omega) + \mathcal{S}_N(\omega)}$$

where  $\mathcal{S}_X(\omega)$  is the power spectrum of the signal and  $\mathcal{S}_N(\omega)$  is the power spectrum of the noise. If this gain can be calculated at the frequencies corresponding to the DFT bins then it can be applied as a noise reduction filter in the DFT domain. If we interchange the continuous frequency variable  $\omega$  with the DFT bin number  $m$  the following ‘pseudo-Wiener’ noise reduction rule is obtained:

$$f(Y(m)) = \frac{\mathcal{S}_X(m)}{\mathcal{S}_X(m) + \mathcal{S}_N(m)} Y(m)$$

where we have dropped the sub-frame variable  $n$  for convenience.

The problem here is that we do not in general know any of the required terms in this equation except for the raw DFT data  $Y(m)$ . However, in many cases it is assumed that  $\mathcal{S}_N(m)$  is known and it only remains to obtain  $\mathcal{S}_X(m)$  ( $\mathcal{S}_N(m)$  can often be estimated from ‘silent’ sections where no music is playing, for example). A very crude estimate for  $\mathcal{S}_Y(m)$ , the power spectrum of the noisy signal, can be obtained from the amplitude squared of the raw DFT components  $|Y(m)|^2$ . An estimate for the signal power spectrum is then:

$$\mathcal{S}_X(m) = \begin{cases} |Y(m)|^2 - \mathcal{S}_N(m), & |Y(m)|^2 > \mathcal{S}_N(m) \\ 0, & \text{otherwise} \end{cases}$$

where the second case ensures that the power spectrum is always greater than or equal to zero, leading to a noise reduction function of the form:

$$f(Y(m)) = \begin{cases} \frac{|Y(m)|^2 - \mathcal{S}_N(m)}{|Y(m)|^2} Y(m), & |Y(m)|^2 > \mathcal{S}_N(m) \\ 0, & \text{otherwise} \end{cases}$$

Defining the power signal-to-noise ratio at each frequency bin to be  $\rho(m) = (|Y(m)|^2 - \mathcal{S}_N(m)) / \mathcal{S}_N(m)$ , this function can be rewritten as:

$$f(Y(m)) = \begin{cases} \frac{\rho(m)}{1 + \rho(m)} Y(m), & \rho(m) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Notice that the noise reducer arising from the Wiener filter introduces zero phase shift - the restored phase equals the phase of the corrupted

signal. Even though it is often quoted that the ear is insensitive to phase, this only applies to time-invariant signals [137]. Thus it can be expected that phase-related modulation effects will be observed when this and other ‘phase-blind’ processes are applied in the time-varying environment of real audio signals.

#### 6.1.1.2 Spectral subtraction and power subtraction

The Wiener solution has the appeal of being based upon a well-defined optimality criterion. Many other criteria are possible, however, including the well known spectral subtraction and power subtraction methods [19, 154, 16, 125, 112].

In the spectral subtraction method an amount of noise equal to the root mean-squared noise in each frequency bin, i.e.  $\mathcal{S}_N(m)^{1/2}$ , is subtracted from the spectral amplitude, while once again the phase is left untouched. In other words we have the following noise reduction function:

$$f(Y(m)) = \begin{cases} \frac{|Y(m)| - \mathcal{S}_N(m)^{1/2}}{|Y(m)|} Y(m), & |Y(m)|^2 > \mathcal{S}_N(m) \\ 0, & \text{otherwise} \end{cases}$$

Rewriting as above in terms of the power signal-to-noise ratio  $\rho(m)$  we obtain:

$$f(Y(m)) = \begin{cases} \left(1 - \frac{1}{(1+\rho(m))^{1/2}}\right) Y(m), & \rho(m) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Another well known variant upon the same theme is the power subtraction method, in which the restored spectral power is set equal to the power of the noisy input minus the expected noise power:

$$f(Y(m)) = \begin{cases} \left(\frac{|Y(m)|^2 - \mathcal{S}_N(m)}{|Y(m)|^2}\right)^{1/2} Y(m), & |Y(m)|^2 > \mathcal{S}_N(m) \\ 0, & \text{otherwise} \end{cases}$$

and in terms of the power signal-to-noise ratio:

$$f(Y(m)) = \begin{cases} \left(\frac{\rho(m)}{1+\rho(m)}\right)^{1/2} Y(m), & \rho(m) > 0 \\ 0, & \text{otherwise} \end{cases}$$

The gain applied here is the square root of the Wiener gain.

The gain curves for Wiener, spectral subtraction and power subtraction methods are plotted in figure 6.2 as a function of  $\rho(m)$ . The output amplitude as a function of input amplitude is plotted in figure 6.3 with  $\mathcal{S}_N(m) = 1$ . It can be seen that spectral subtraction gives the most severe suppression of spectral amplitudes, power subtraction the least severe and

the Wiener gain lies between the two. The Wiener and power subtraction rules tend much more rapidly towards unity gain at high input amplitudes. In our experience the audible differences between the three rules are quite subtle for audio signals, especially when some of the modifications outlined below are incorporated, and the Wiener rule is usually an adequate compromise.

The de-noising procedure is now illustrated with a synthetic example in which a music signal (figure 6.4(a)) is artificially corrupted with additive white Gaussian noise (figure 6.4(b)). Figure 6.4(c) shows the noisy input after pre-windowing with a Hanning window. The data is fast Fourier transformed (FFT-ed) to give figure 6.5 in which the clean data FFT (figure 6.5(a)) is included for comparison. Note the way in which the noise floor has been elevated in the noisy case (figure 6.5(b)). Only the first 200 frequency bins have been plotted, a frequency range of roughly 8.5kHz, as there is little signal information to be seen at higher frequencies for this example. Figure 6.6 shows the filter gain calculated according to the three criteria described above. Note the more severe attenuating effect of the spectral subtraction method. This can also be seen in the reconstructed outputs of figure 6.7, in which there is clearly a progression in the amount of residual noise left in the three methods (see the next section for a discussion of residual noise effects).

### 6.1.2 *Artefacts and ‘musical noise’*

The above methods can all lead to significant reduction in background noise in audio recordings. However, there are several significant drawbacks which inhibit the practical application of these techniques without further modification. The main drawbacks are in the residual noise artefacts, the most annoying of which is known variously as ‘musical noise’, ‘bird-song’ or ‘tonal noise’. We will describe this effect and suggest some ways to eliminate it, noting that attempts to remove these artifacts are likely to involve a trade-off with the introduction of other distortions into the audio.

Musical noise arises from the randomness inherent in the crude estimate for the signal power spectrum which is used in the basic form of these methods, i.e.  $\mathcal{S}_X(m) = |Y(m)|^2 - \mathcal{S}_N(m)$ . Clearly the use of the raw input spectral amplitude will lead to inaccuracies in the filter gain owing to the random noise and signal fluctuations within each frequency band. This can lead to over-estimates for  $\mathcal{S}_X(m)$  in cases where  $|Y(m)|$  is too high and under-estimates when  $|Y(m)|$  is too low, and a corresponding random distortion of the sound quality in the restored output. The generation of musical noise can be seen most clearly in the case where no signal is present ( $\mathcal{S}_X(m) = 0$ ) and the filter gain would ideally then be zero across all frequency bins. To see what actually occurs we use another synthetic example, in which a pure white Gaussian noise signal is de-noised using the Wiener suppression rule. In order to illustrate the effect best the noise power spec-

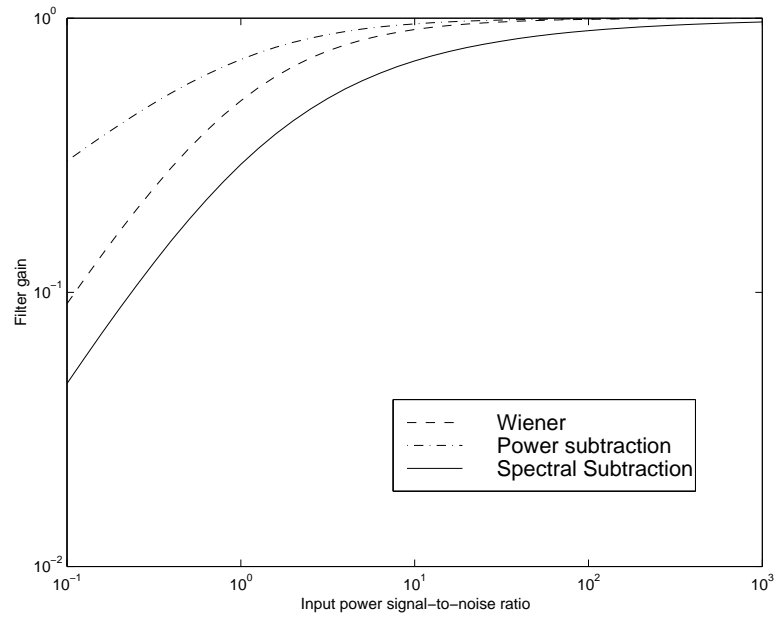


FIGURE 6.2. Filter gain at frequency bin  $m$  as a function of power signal-to-noise ratio  $\rho(m)$ .

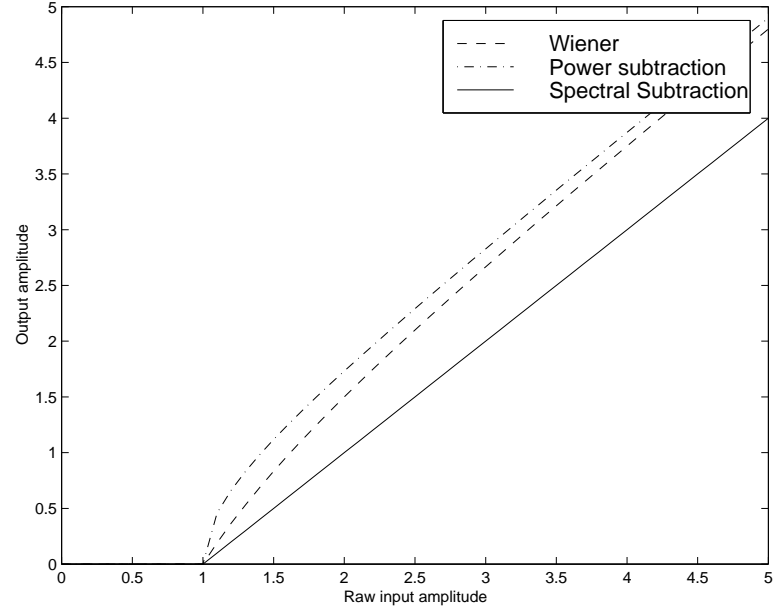


FIGURE 6.3. Filter output amplitude at frequency bin  $m$  as a function of input amplitude  $|Y(m)|$ .



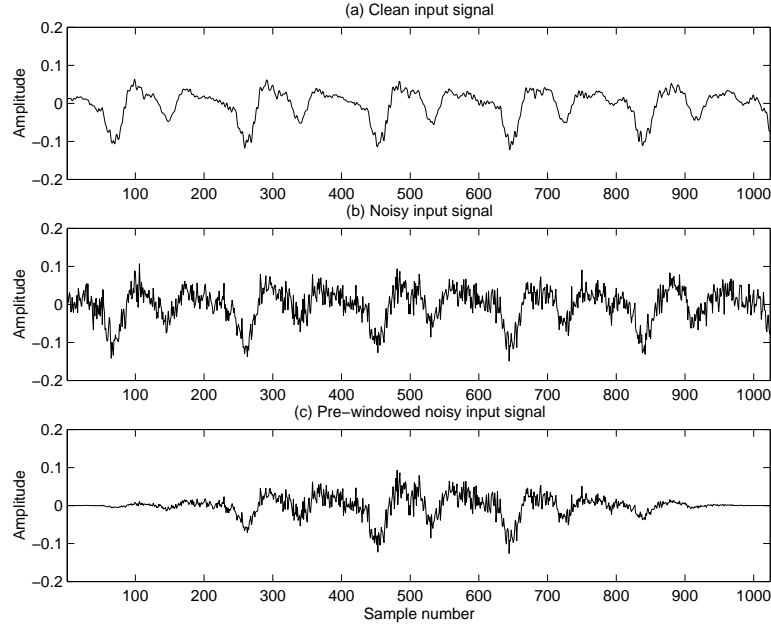


FIGURE 6.4. (a) Clean input signal  $x_n$ , (b) Noisy input signal  $y_n$ , (c) Pre-windowed input  $gy_n$ .

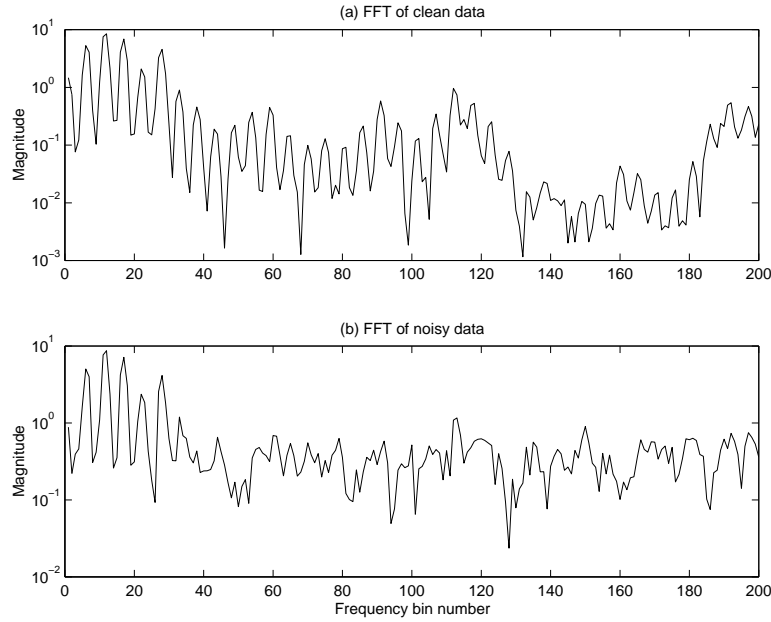


FIGURE 6.5. (a) FFT of clean data  $|X(m)|$ , (b) FFT of noisy data  $|Y(m)|$ .

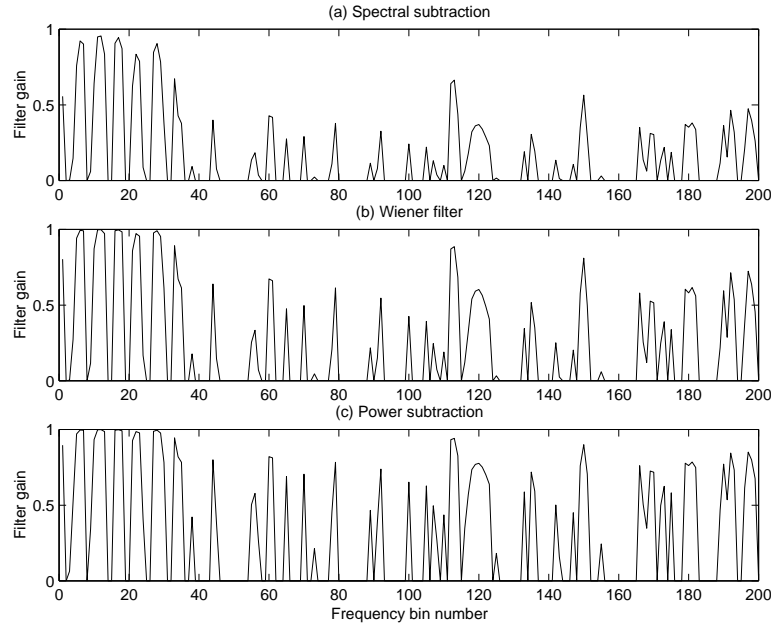


FIGURE 6.6. Filter gain for (a) Spectral subtraction, (b) Wiener filter, (c) Power subtraction.

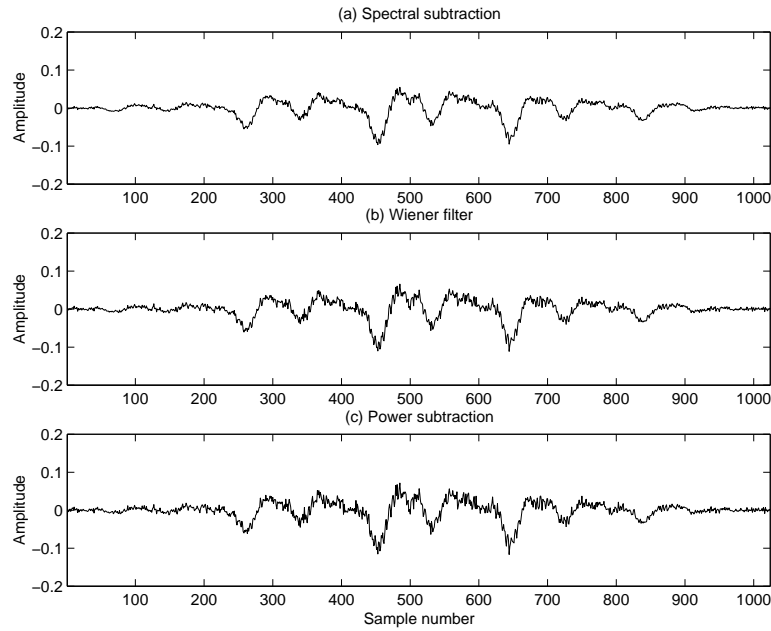


FIGURE 6.7. Estimated signal prior to post-windowing and overlap-add resynthesis: (a) Spectral subtraction, (b) Wiener filter, (c) Power subtraction.

trum is over-estimated by a factor of two, which leads to a smaller amount of residual noise but greater distortion to the musical signal components. Figure 6.8 shows the input noise signal, the pre-windowed noise signal and the Wiener estimated signal corresponding to this noise signal. The noise is well attenuated (roughly 10dB of attenuation can be measured) but as before a residual noise is clearly visible. Ideally we would like the residual noise to have a pleasant-sounding time-invariant characteristic. However, figure 6.9 shows that this is not the case. Figure 6.9(a) shows the spectral magnitude of the noise input signal and figure 6.9(b) gives the corresponding noise reduced spectrum. It can be seen that while low amplitude noise components have been attenuated to zero, as desired, the high amplitude components have remained largely unsuppressed, owing to the non-linear nature of the suppression rule. These components are typically isolated in the spectrum from one another and will thus be perceived as ‘tones’ in the restored output. When considered from frame to frame in the data these isolated components will move randomly around the spectrum, leading to rapidly time-varying bursts of tonal or ‘musical’ noise in the restoration. This residual can be more unpleasant than the original unattenuated noise, and we now go on to discuss ways for alleviating the problem.

### 6.1.3 Improving spectral domain methods

We have already stated that the main cause of musical noise is random fluctuations in the estimate of  $\mathcal{S}_X(m)$ , the signal power spectrum. Clearly one way to improve the situation will be to make a more statistically stable estimate of this term for each frame. Another way to improve the quality of restoration will be to devise alternative noise suppression rules based upon more appropriate criteria than those described above. There is a large number of techniques which have been proposed for achieving either or both of these objectives, mostly aimed at speech enhancement, and we do not attempt to describe all of them here. Rather we describe the main principles involved and list some of the key references. For a range of audio applications based upon these techniques and their variants the reader is referred to [109, 138, 187, 190, 189, 185, 58, 116].

#### 6.1.3.1 Eliminating musical noise

The simplest way to eliminate musical noise is to over-estimate the noise power spectrum, see e.g. [187, 190, 16, 17, 116], in a similar (but more extreme) way as used in the results of figure 6.8. To achieve this we simply replace  $\mathcal{S}_N(m)$  with  $\alpha\mathcal{S}_N(m)$ , where  $\alpha > 1$  is typically between 3 and 6 for noisy audio. Of course, such a procedure simply trades off improvements in musical noise with distortion to the musical signal. Another simple idea leaves a noise ‘floor’ in the spectrum, masking the isolated noise peaks which lead to tonal noise. The introduction of a noise floor means less noise

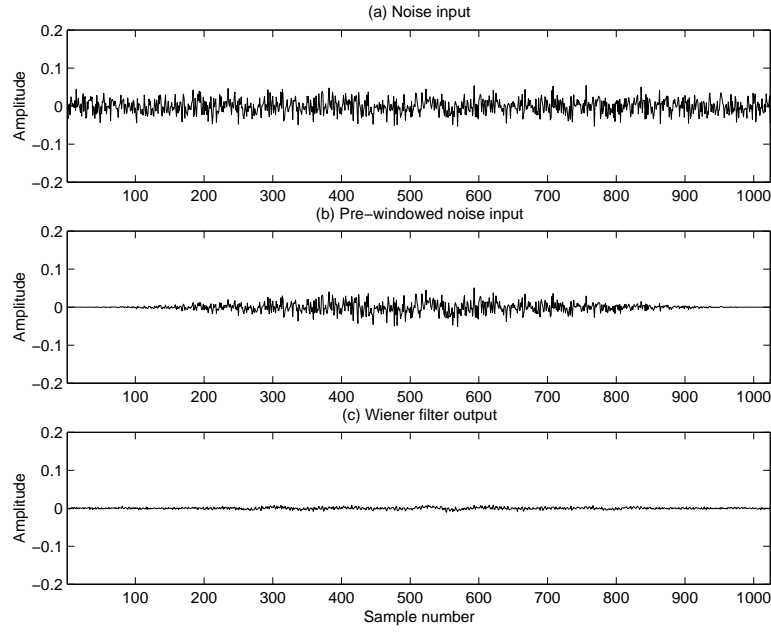


FIGURE 6.8. (a) Input noise signal  $y_n$ , (b) Pre-windowed input  $g_l y_n$ , (c) Wiener estimated output signal.

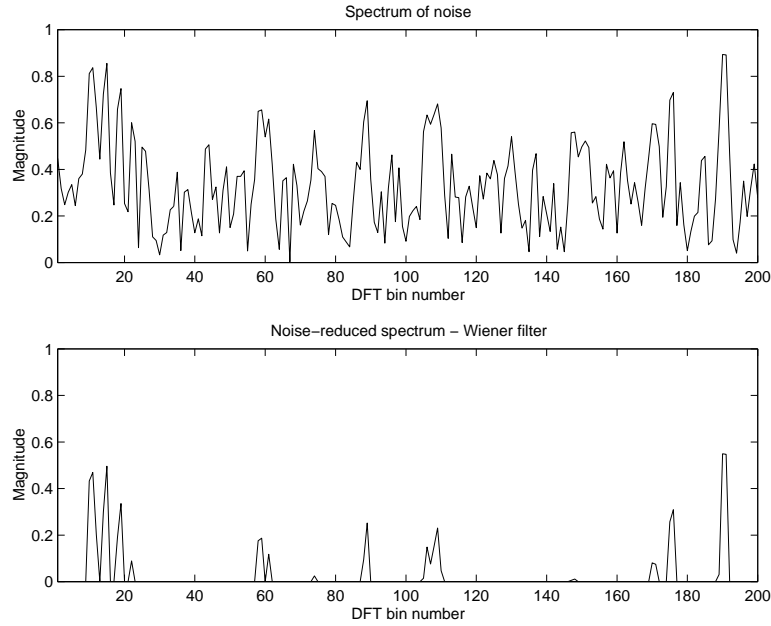


FIGURE 6.9. (a) FFT of noisy data  $|Y(m)|$  (b) FFT of noise reduced data  $|f(Y(m))|$ .

reduction, but this might be acceptable depending upon the application and noise levels in the original recording. A first means of achieving this [16] simply limits the restored spectral amplitude to be no less than  $\beta S_N(m)^{1/2}$ , where  $0 < \beta < 1$  determines the required noise attenuation. The resulting residual sounds slightly unnatural, and a better scheme places a lower limit on the filter gain [16, 17, 116]. The Wiener suppression rule would become, for example:

$$f(Y(m)) = \max \left( \alpha, \frac{\rho(m)}{1 + \rho(m)} \right) Y(m)$$

This achieves a significant degree of masking for musical noise and leaves a natural sounding residual. It can also help to limit any loss of sound quality by ensuring that very small signal components are never completely attenuated.

The best means for elimination of musical noise to date, however, employ temporal information from surrounding data sub-frames. It is highly unlikely that a random noise peak will occur at the same frequency in adjacent sub-frames, while tonal components which are genuinely present in the music are likely to remain at the same frequency for several sub-frames. Thus some degree of linear or non-linear processing can be used to eliminate the noise spikes while retaining the more slowly varying signal components intact. In [19] a spectral magnitude averaging approach is suggested in which the raw speech amplitudes  $|Y(n, m)|$  are averaged over several adjacent sub-frames  $n$  at each frequency bin  $m$ . This helps somewhat in the stability of signal power spectral estimates, but will smooth out genuine transients in the music without leading to a very significant reduction in musical noise. A more successful approach from the same paper uses the minimum estimated spectral amplitude from adjacent sub-frames as the restored output, although this can again lead to a ‘dulling’ of the sound quality. One simple technique which we have found to be effective without any serious degradation of the sound quality involves taking a median of adjacent blocks rather than an arithmetic mean or minimum value. This eliminates musical noise quite successfully and the effect on the signal quality is less severe than taking a minimum amplitude. Other techniques include non-linear ‘masking’ out of frequency components whose temporal neighbours are of low amplitude (and hence can be classified as noise) [189] and applying frequency domain de-noising to the temporal envelope of individual frequency bins [29].

In the same way that there are many possible methods for performing click detection, there is no end to the alternative schemes which can be devised for refining the hiss reduction performance based upon temporal spectral envelopes. In practice a combination of several techniques and some careful algorithmic tuning will be necessary to achieve the best performance.

### 6.1.3.2 Advanced noise reduction functions

In addition to the basic noise suppression rules (Wiener, power/spectral subtraction) described so far, various alternative rules have been proposed, based upon criteria such as maximum likelihood [125] and minimum mean-squared error [54, 55, 56]. In [125] a maximum likelihood estimator is derived for the spectral amplitude components under Gaussian noise assumptions, with the complex phase integrated out. This estimator is extended in [55] to the minimum mean-square error estimate for the amplitudes, which incorporates Gaussian *a priori* assumptions for the signal components in addition to the noise. Both of these methods [125, 55] also incorporate uncertainty about the *presence* of signal components at a given frequency, but rely on the assumption of statistical independence between spectral components at differing frequencies. In fact, the Ephraim and Malah method [55] is well known in that it does not generate significant levels of musical noise. This can be attributed to a time domain non-linear smoothing filter which is used to estimate the *a priori* signal variance [31, 84].

### 6.1.3.3 Psychoacoustical methods

A potentially important development in spectral domain noise reduction is the incorporation of the psychoacoustical properties of human hearing. It is clearly sub-optimal to design algorithms based upon mathematical criteria which do not account for the properties of the listener. In [27, 28] a method is derived which attempts to mimic the pre-filtering of the auditory system for hearing-impaired listeners, while in [183, 116] simultaneous masking results of the human auditory system [137] are employed to predict which parts of the spectrum need not be processed, hence leading to improved fidelity in the restored output as perceived by the listener. These methods are in their infancy and do not yet incorporate other aspects of the human auditory system such as non-simultaneous masking, but it might be expected that noise reducers and restoration systems of the future will take fuller account of these features to their benefit.

### 6.1.4 Other transform domain methods

So far we have assumed a Fourier transform-based implementation of spectral noise reduction methods. It is possible to apply many of these methods in other domains which may be better tuned to the non-stationary and transient character of audio signals. Recent work has seen noise reduction performed in alternative basis expansions, in particular the wavelet domain [134, 185, 14] and sub-space representations [44, 57].

## 6.2 Model-based methods

The spectral domain methods described so far are essentially non-parametric, although some include assumptions about the probability distribution of signal and noise spectral components. In the same way as for click reduction, it should be possible to achieve better results if an appropriate signal model can be incorporated into the noise reduction procedure, however noise reduction is an especially sensitive operation and the model must be chosen very carefully. In the speech processing field Lim and Oppenheim [111] studied noise reduction using an autoregressive signal model, deriving iterative MAP and ML procedures. These methods are computationally intensive, although the signal estimation part of the iteration is shown to have a simple frequency-domain Wiener filtering interpretation (see also [147, 107] for Kalman filtering realisations of the signal estimation step). It is felt that new and more sophisticated model-based procedures may provide noise reducers which are competitive with the well-known short-time Fourier based methods. In particular, modern Bayesian statistical methodology for solution of complex problems [172, 68] allows for more realistic signal and noise modelling, including non-Gaussianity, non-linearity and non-stationarity. Such a framework can also be used to perform joint restoration of both clicks and random noise in one single process. A Bayesian approach to this joint problem using an autoregressive signal model is described in [70] and in chapter 9 (the ‘noisy data’ case), while [141, 142] present an extended Kalman filter for ARMA-modelled audio signals with the same type of degradation. See also [81, 72, 73] for some recent work in this area which uses Markov chain Monte Carlo (MCMC) methods to perform joint removal of impulses and background noise for both AR and autoregressive moving-average (ARMA) signals. Chapter 12 describes in detail the principles behind these methods. The standard time series models have yielded some results of a good quality, but it is anticipated that more sophisticated and realistic models will have to be incorporated in order to exploit the model-based approach fully.

### 6.2.1 Discussion

A number of noise reduction methods have been described, with particular emphasis on the short-term spectral methods which have been most popular to date. However, it is expected that new statistical methodology and rapid increases in readily-available computational power will lead in the future to the use of more sophisticated methods based on realistic signal modelling assumptions and perceptual optimality criteria.





# Part III

## Advanced Topics



# 7

## Removal of Low Frequency Noise Pulses

A problem which is common to several recording media, including gramophone discs and optical film sound tracks, is that of low frequency noise pulses. This form of degradation is typically associated with large scratches or even breakages in the surface of a gramophone disc. The precise form of the noise pulses depends upon the characteristics of the playback system but a typical waveform is shown in figure 7.1. A large discontinuity is observed followed by a decaying low frequency transient. The noise pulses appear to be additively superimposed upon the undistorted signal waveform (see figure 7.2).

In this chapter we briefly review some early digital technology for removal of these defects, the templating method [187, 190], noting that problems can arise with these methods since they rely on little or no variability of pulse waveforms throughout a recording. This can be quite a good assumption when defects are caused by single breakages or radial scratches on a disc recording. However pulse shapes are much less consistent when defects are caused by randomly placed scratches in disc recordings or optical film sound tracks. The templating method then becomes very difficult to apply. Many different templates are required to cover all possible pulse shapes, and automation of such a system is not straightforward. Thus it seems better to form a more general model for the noise pulses which is sufficiently flexible to allow for all the pulses encountered on a particular recording. In this chapter we propose the use of an AR model for the noise pulses, driven by impulsive excitation. An algorithm is initially derived for the separation of additively superimposed AR signals. This is then modified to

give a restoration algorithm suited to our assumed models for signal and noise pulses.

## 7.1 Existing methods

Low frequency noise pulses appear to be the response of the pick-up system to extreme step-like or impulsive stimuli caused by breakages in the groove walls of gramophone discs or large scratches on an optical film sound track. The audible effect of this response is a percussive ‘pop’ or ‘thump’ in the recording. This type of degradation is often the most disturbing artefact present in a given extract. It is thus highly desirable to eliminate noise pulses as a first stage in the restoration process.

Since the majority of the noise pulse is of very low frequency it might be thought that some kind of high pass filtering operation would remove the defect. Unfortunately this does not work well since the discontinuity at the front of the pulse has significant high frequency content. Some success has been achieved with a combination of localised high pass filtering and interpolation to remove discontinuities. However it is generally found that significant artefacts remain after processing or that the low frequency content of the signal has been damaged.

The only existing method known to the authors is that of Vaseghi and Rayner [187, 190]. This technique, which employs a template for the noise pulse waveform, has been found to give good results for many examples of broken gramophone discs. The observation was made that the low frequency sections of successive occurrences of noise pulses in the same recording were nearly identical in envelope (to within a scale factor). This observation is true particularly when the noise pulses correspond to a single fracture running through successive grooves of a disc recording. Given the waveform of the repetitive section of the noise pulse (the ‘template’  $t_m$ ) it is then possible to subtract appropriately scaled versions from the corrupted signal  $y_n$  wherever pulses are detected. Any remaining samples close to the discontinuity which are irrevocably distorted can then be interpolated using a method such as the LSAR interpolator discussed earlier.

The template  $t_m$  is obtained by long term averaging of many such pulses in the corrupted waveform. The correct position and scaling for each individual pulse are obtained by cross-correlating the template with the corrupted signal.

The template method is limited in several important ways which prevent automation of pulse removal. While the assumption of constant template shape is good for short extracts with periodically recurring noise pulses (e.g. in the case of a broken gramophone disc) it is not a good assumption for many other recordings. Even where noise pulses do correspond to a single radial scratch or fracture on the record the pulse shape is often

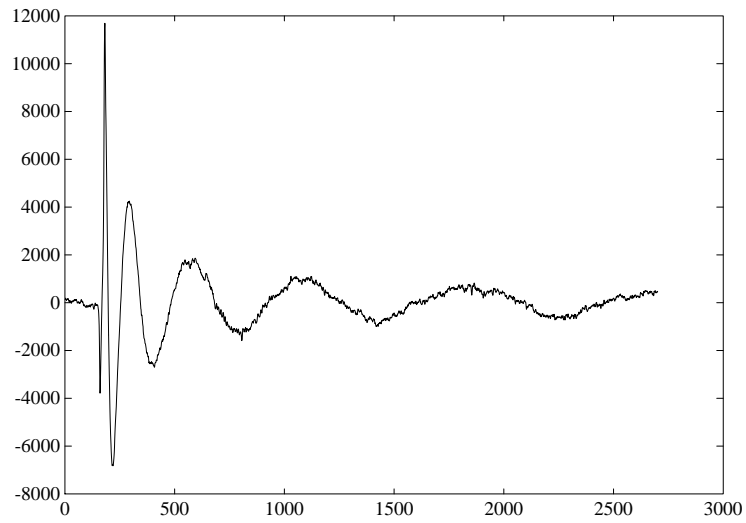


FIGURE 7.1. Noise pulse from broken gramophone disc ('lead-in' groove)

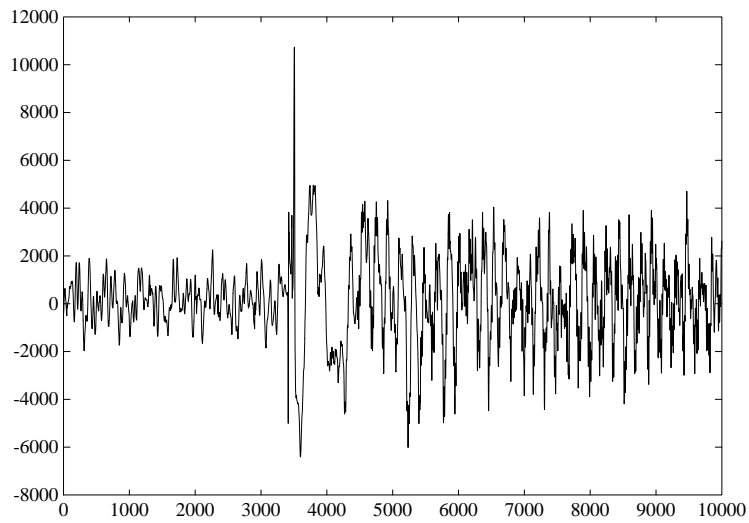


FIGURE 7.2. Signal waveform degraded by additive noise pulse

found to change significantly as the recording proceeds, while much more variety is found where pulses correspond to ‘randomly’ placed scratches and breakages on the recording. Further complications arise where several pulses become superimposed as is the case for several closely spaced scratches.

Correct detection is also a problem. This may seem surprising since the defect is often very large relative to the signal. However, audible noise pulses do occur in high amplitude sections of the signal. In such cases the cross-correlation method of detection can give false alarms from low frequency components in the signal; in other circumstances noise pulses can be missed altogether. This is partly as a result of colouration of the signal which renders the cross-correlation procedure sub-optimal. A true matched filter for the noise pulse would take into account the signal correlations (see e.g. [186]) and perhaps achieve some improvements in detection. However this issue is not addressed here since the other limitations of the templating method are considered too severe. Rather in chapter 7 we derive a more general restoration system for noise pulses which is designed to cope with situations where pulses are of varying shape and at random locations. With such an approach we aim to restore a wider range of degraded material, including optical sound tracks, with little or no user intervention.

## 7.2 Separation of AR processes

Consider firstly the problem of separating two additively superimposed AR processes. This will provide the basis for the subsequent noise pulse restoration algorithm. A probabilistic derivation is given here for consistency with the remainder of the book, although a least squares analysis yields similar results.

A data block  $\mathbf{y}$  with  $N$  samples is expressed as the sum of two constituent parts  $\mathbf{x}_1$  and  $\mathbf{x}_2$ :

$$\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2. \quad (7.1)$$

We wish to estimate either  $\mathbf{x}_1$  or  $\mathbf{x}_2$  from the combined signal  $\mathbf{y}$  under the assumption that both constituent signals are drawn from AR processes. The parameters and excitation variances  $\{\mathbf{a}_1, \sigma_{e1}^2\}$  and  $\{\mathbf{a}_2, \sigma_{e2}^2\}$  for the two AR processes are assumed known. The two signals  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are assumed independent and their individual PDFs can be written in the usual fashion (see (4.54)) under the Gaussian independence assumption as

$$p_{\mathbf{x}_1}(\mathbf{x}_1) = \frac{1}{(2\pi\sigma_{e1}^2)^{\frac{N-P_1}{2}}} \exp\left(-\frac{1}{2\sigma_{e1}^2} \mathbf{x}_1^T \mathbf{A}_1^T \mathbf{A}_1 \mathbf{x}_1\right) \quad (7.2)$$

$$p_{\mathbf{x}_2}(\mathbf{x}_2) = \frac{1}{(2\pi\sigma_{e2}^2)^{\frac{N-P_2}{2}}} \exp\left(-\frac{1}{2\sigma_{e2}^2} \mathbf{x}_2^T \mathbf{A}_2^T \mathbf{A}_2 \mathbf{x}_2\right) \quad (7.3)$$

where  $P_i$  is the AR model order for  $i$ th process ( $i = 1, 2$ ) and  $\mathbf{A}_i$  is the prediction matrix containing AR model coefficients from the  $i$ th model. These PDFs are, as before, strictly conditional upon the first  $P_i$  data samples for each signal component. The resulting approximation is maintained for the sake of notational simplicity. Note, however, that as before the exact likelihoods can be incorporated by relatively straightforward modifications to the first  $P_i$  elements of the first  $P_i$  rows of the matrices  $\mathbf{A}_i^T \mathbf{A}_i$  (see appendix C).

Say we wish to find the MAP solution for  $\mathbf{x}_1$ . The probability for the combined data conditional upon the first component signal is given by:

$$p(\mathbf{y} \mid \mathbf{x}_1) = p_{\mathbf{x}_2}(\mathbf{y} - \mathbf{x}_1). \quad (7.4)$$

We then obtain the posterior distribution for  $\mathbf{x}_1$  using Bayes' theorem as

$$\begin{aligned} p(\mathbf{x}_1 \mid \mathbf{y}) &\propto p(\mathbf{y} \mid \mathbf{x}_1) p_{\mathbf{x}_1}(\mathbf{x}_1) \\ &\propto p_{\mathbf{x}_2}(\mathbf{y} - \mathbf{x}_1) p_{\mathbf{x}_1}(\mathbf{x}_1) \end{aligned} \quad (7.5)$$

where the term  $p(\mathbf{y})$  can as usual be ignored as a normalising constant.

Substituting for the terms in (7.5) using (7.2) and (7.3) then gives

$$p(\mathbf{x}_1 \mid \mathbf{y}) \propto \frac{\exp\left(-\frac{1}{2\sigma_{e2}^2}(\mathbf{y} - \mathbf{x}_1)^T \mathbf{A}_2^T \mathbf{A}_2 (\mathbf{y} - \mathbf{x}_1) - \frac{1}{2\sigma_{e1}^2} \mathbf{x}_1^T \mathbf{A}_1^T \mathbf{A}_1 \mathbf{x}_1\right)}{(2\pi\sigma_{e1}^2)^{(N-P_1)/2} (2\pi\sigma_{e2}^2)^{(N-P_2)/2}} \quad (7.6)$$

and the MAP solution vector  $\mathbf{x}_1^{\text{MAP}}$  is then obtained as the solution of :

$$\left( \frac{\mathbf{A}_1^T \mathbf{A}_1}{\sigma_{e1}^2} + \frac{\mathbf{A}_2^T \mathbf{A}_2}{\sigma_{e2}^2} \right) \mathbf{x}_1^{\text{MAP}} = \frac{\mathbf{A}_2^T \mathbf{A}_2}{\sigma_{e2}^2} \mathbf{y}. \quad (7.7)$$

This separation equation requires knowledge of both AR processes and as such cannot straightforwardly be applied in general 'blind' signal separation applications. Nevertheless we will now show that a modified form of this approach can be applied successfully to the practical case of separating an audio signal from transient noise pulses.

### 7.3 Restoration of transient noise pulses

Examination of figures 7.1 and 7.2 indicates that the noise pulse is composed of a high amplitude impulsive start-up transient followed by a low frequency decaying oscillatory tail consistent with a 'free-running' system (i.e. a resonant system with no excitation). The oscillatory tail appears to be additive to the true audio signal while the impulsive start may well obliterate the underlying signal altogether for a period of some large number of samples (50-100 samples is typical).

This noise transient is interpreted as being the response of the mechanical playback apparatus to highly impulsive or step-like stimuli such as would occur when a complete breakage or deep gouge is encountered by the stylus in the groove wall. The way such discontinuities are generated, in contrast with the way the cutter records the true audio signal onto the disc, means that the damaged grooves can have a much higher rate of rise or fall than could ever be encountered from the standard recording procedure (and most likely at a higher rate of change than the subsequent electronic circuitry is capable of reproducing). It is postulated that these especially sharp step-like discontinuities excite the low frequency resonance observed with some scratches, while other clicks and scratches with less sharp transitions will not excite the same degree of resonance. The frequency of oscillation is consistent with the resonant frequency of the tone arm apparatus (15-30Hz, see [133]) so this may well be the source of the resonance. We do not verify this here from mechanics theory or experiments but rather concern ourselves with the modelling of the noise transients in the observed waveform.

It can be seen clearly from figure 7.1 that the frequency of resonance decreases significantly as the transient proceeds. This is typical of noise pulses from both gramophone recordings and optical film sound tracks. Attempts to model the pulse as several superimposed decaying vibration modes with different frequencies have proved largely unsuccessful, and it seems more likely that there is non-linearity in the mechanical system which would account for this effect. This kind of change in resonant frequency is consistent with a ‘stiffening’ spring system in which a spring-type mechanical element has increasing stiffness with increasing displacement (see e.g. [90]). Exploratory experiments indicate that discrete low order (2-3) non-linear AR models based on a reduced Volterra expansion (see e.g. [18]) lead to significantly better whitening of the prediction error than a linear AR model with the same number of coefficients. This requires further investigation and is left as further work. Here, for the sake of achieving analytic results (and hence a practical restoration system), we adopt the linear AR model as an approximation to the true mechanical system. The impulsive start-up section is modelled as high-amplitude impulsive input to the system and a low level white noise excitation is assumed elsewhere to allow for modelling inaccuracies resulting from the linear approximation.

### 7.3.1 *Modified separation algorithm*

The proposed model for noise transient degradation is similar to the ‘noisy data’ model for standard click degradation (see section 9.1) in which a noise signal is generated by switching between two noise processes  $\{n_m^1\}$  and  $\{n_m^0\}$  according to the value of a binary process  $\{i_m\}$ . The high amplitude noise process  $\{n_m^1\}$  corresponds to high level impulsive input (i.e. the start-up transient) while the low level noise process  $n_m^0$  represents the low level of noise excitation mentioned above. The low frequency transient model



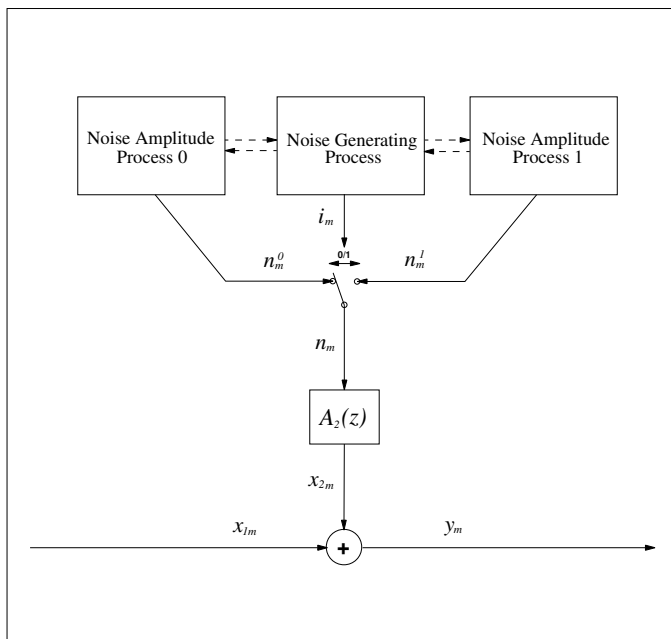


FIGURE 7.3. Schematic model for noise transients

differs from the standard click model in that the switched noise process is then treated as the excitation to the AR process with parameters  $\mathbf{a}_2$ . The output of this process is then added to the input data signal. Figure 7.3 gives a schematic form for the noise model. Since this model does not specify when the high amplitude impulsive excitation process  $n_m^1$  is active, it is general enough to include the problematic scenario of many large scratches close together, causing ‘overlapping’ noise pulses in the degraded audio waveform. We now consider modifications to the AR-based separation algorithm given in the previous section. These will lead to an algorithm for restoring signals degraded by noise from the proposed model.

In the simple AR-based separation algorithm (7.7) a constant variance Gaussian white excitation was assumed for both AR processes. However, the model discussed above indicates a switched process for the excitation to AR process  $\mathbf{a}_2$ , depending on the current ‘switch value’  $i_m$ . We define a vector  $\mathbf{i}$  similar to the detection vector of chapter 9 which contains a ‘1’ wherever  $i_m = 1$ , i.e. the high amplitude noise process is active, and ‘0’ otherwise. This vector gives the switch position at each sampling instant and hence determines which of the two noise processes is applied to the AR filter  $\mathbf{a}_2$ . If the two noise processes are modelled as Gaussian and white with variances  $\sigma_{n0}^2$  and  $\sigma_{n1}^2$  the correlation matrix for the excitation sequence

to the second AR model  $\mathbf{a}_2$  is a diagonal matrix  $\Lambda$ . The diagonal elements  $\lambda_m$  of  $\Lambda$  are given by

$$\lambda_m = \sigma_{n0}^2 + i_m(\sigma_{n1}^2 - \sigma_{n0}^2). \quad (7.8)$$

The PDF for  $\mathbf{x}_2$ , the ‘noise’ signal, is then modified from the expression of (7.3) to give

$$p_{\mathbf{x}_2}(\mathbf{x}_2) = \frac{1}{(2\pi)^{\frac{N-P_2}{2}} |\Lambda|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{x}_2^T \mathbf{A}_2^T \Lambda^{-1} \mathbf{A}_2 \mathbf{x}_2\right). \quad (7.9)$$

The derivation for the modified separation algorithm then follows the same steps as for the straightforward AR separation case, substituting the modified PDF for  $\mathbf{x}_2$  and leading to the following posterior distribution,

$$p(\mathbf{x}_1 | \mathbf{y}) \propto \frac{\exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{x}_1)^T \mathbf{A}_2^T \Lambda^{-1} \mathbf{A}_2 (\mathbf{y} - \mathbf{x}_1) - \frac{1}{2\sigma_{e1}^2} \mathbf{x}_1^T \mathbf{A}_1^T \mathbf{A}_1 \mathbf{x}_1\right)}{(2\pi\sigma_{e1}^2)^{(N-P_1)/2} (2\pi)^{(N-P_2)/2} |\Lambda|^{1/2}} \quad (7.10)$$

and the resulting MAP solution vector  $\mathbf{x}_1^{\text{MAP}}$  is obtained as the solution of:

$$\left(\frac{\mathbf{A}_1^T \mathbf{A}_1}{\sigma_{e1}^2} + \mathbf{A}_2^T \Lambda^{-1} \mathbf{A}_2\right) \mathbf{x}_1^{\text{MAP}} = \mathbf{A}_2^T \Lambda^{-1} \mathbf{A}_2 \mathbf{y} \quad (7.11)$$

If  $\sigma_{n1}^2$  is very large the process  $\{n_m^1\}$  tends to a uniform distribution, in which case the impulsive section will largely obscure the underlying audio waveform  $\mathbf{x}_1$ . In the limit the elements of  $\Lambda^{-1}$  equal to  $(1/\sigma_{n1}^2)$  tend to zero. Since  $\sigma_{n1}^2$  is usually large and awkward to estimate we will assume the limiting case for the restoration examples given later.

The algorithm as it stands is appropriate for restoration of a whole block of data with no reference to the surroundings. Discontinuities can thus result at the boundaries between a restored data block and adjacent data. Continuity can be enforced with a windowed overlap/add approach at the ends of a processed block. However, improved results have been obtained by fixing a set of known samples at both ends of the block. Continuity is then inherent in the separation stage. This approach requires a slightly modified version of equation (7.11) which accounts for the known samples in a manner very similar to the standard AR-based interpolator (see section 5.2.2) in which known samples are fixed either side of missing data.

### 7.3.2 Practical considerations

The modified algorithm proposed for separation of the true signal from the combination of true signal and noise transients requires knowledge of both AR systems, including noise variances and actual parameter values,

as well as the detection vector  $\mathbf{i}$  which indicates which noise samples have impulsive excitation and which have low level excitation. We now consider how we might obtain estimates for these unknowns and hence produce a practical restoration scheme.

### 7.3.2.1 Detection vector $\mathbf{i}$

Identification of  $\mathbf{i}$  is the equivalent for noise transients of click detection. We are required to estimate which samples of the noise pulse signal correspond to impulsive noise excitation. A Bayesian scheme for optimal detection can be derived based on the same principles as the click detector of chapter 9 but using the revised noise model. However, in many cases the noise transients are large relative to the audio signal and a simpler approach will suffice.

An approach based on the inverse filtering AR detector as described elsewhere (see section 5.3.1) has been found to perform adequately in most cases. The impulsive sections of the low frequency noise transients are simply considered as standard clicks of high amplitude. The inverse filtering threshold operates as for detection of standard clicks, but a second higher valued threshold is chosen to ‘flag’ clicks of the low frequency transient type. These clicks are then restored using the scheme introduced above. Such a detection procedure will provide good results for most types of noise pulse degradation. However, ‘false alarms’ and ‘missed detections’ are an inevitable consequence in a signal where low frequency transients can be of small amplitude or when standard clicks can be of comparable magnitude to noise transient clicks. The higher threshold must be chosen from trial and error and will reflect how tolerant we are of false alarms (which may result in distortion of the low frequency signal content) and missed detections (which will leave small transients unrestored). The more sophisticated Bayesian detection scheme should improve the situation but is left as a topic for future investigation.

### 7.3.2.2 AR process for true signal $\mathbf{x}_1$

The AR model parameters and excitation variance  $\{\mathbf{a}_1, \sigma_{e1}^2\}$  for the ungraded audio source can usually be estimated by standard methods (see section 4.3.1) from a section of uncorrupted data in the close vicinity of the noise pulse. It is best to estimate these parameters from data prior to the noise pulse so as to avoid the effects of the low frequency noise tail on estimates. Assumptions of local stationarity are then made for restoration of the subsequent corrupted section of data using the same AR model. The AR model order is fixed beforehand and, as for click removal applications, an order of 40-60 is typically sufficient.

### 7.3.2.3 AR process for noise transient $\mathbf{x}_2$

The AR process for the noise transients may be identified in several ways. Perhaps the most effective method is to estimate the parameters by maximum likelihood from a ‘clean’ noise pulse obtained from a passage of the recording where no recorded sound is present, such as the lead-in groove of a disc recording (see figure 7.1). If a ‘clean’ noise transient is not available for any reason it is possible to use estimates obtained from another similar recording. These estimates can of course be replaced or adapted based on restored pulses obtained as processing proceeds. Typically a very low model order of 2-3 is found to perform well. When no such estimates are available we have found that a ‘second differencing’ AR model, which favours signals which are ‘smooth’ in terms of their second differences, with parameters  $[2, -1]$ , leads to restorations of a satisfactory standard.

### 7.3.3 *Experimental evaluation*

Experimental results are now presented for the signal separation algorithms derived in previous sections. Firstly we consider a synthetic case in which a noise transient obtained from the lead-in groove of a gramophone recording is added at a known position to a ‘clean’ audio waveform. The resulting synthetically corrupted waveform is shown in figure 7.4. The restoration corresponding to this waveform is shown below in figure 7.5 and we can see that the degradation is completely removed as far as the eye can see. The next two figures (7.6 and 7.7) show the true noise transient and the transient estimated by the algorithm, respectively. The estimated transient is obtained by subtracting the restored waveform (figure 7.5) from the corrupted waveform (figure 7.4). There is very good correspondence between the true and estimated transient waveform, indicating that the separation algorithm has worked well. We now consider the procedure in more detail.

The AR model parameters for the noise transient were estimated to order 2 by the covariance method from the non-impulsive section of the true noise transient (samples 500-1000 of figure 7.6). The AR parameters for the signal were estimated to order 80 from a section of 1000 data points just prior to the noise transient. The high amplitude impulsive section driving the noise transient was chosen as a run of 50 samples from 451-500 and the separation algorithm operated over the larger interval 441-720.

In this example 80 ‘restored’ samples were fixed before and after the region of signal separation (samples 441-720). The values of these fixed samples were obtained by a high-pass filtering operation on the corrupted data. Such a procedure assumes little or no spectral overlap between the desired signal and noise transient at large distances from the impulsive central section and leads to computational savings, since a significant proportion of the transient can be removed by standard high-pass filtering operations.

The estimated noise transient under these conditions is shown in more detail in figure 7.9. The region of signal separation is enclosed by the vertical dot-dash line while the region of high amplitude impulsive input is enclosed by the dotted lines. For comparison purposes the true noise pulse is shown in figure 7.8. A very close correspondence can be seen over the whole region of separation although the peak values are very slightly under-predicted. Figures 7.10 and 7.11 show the same region in the true and estimated audio signals. Once again a very good correspondence can be observed.

Two further examples are now given which illustrate restoration performed on genuine degraded audio material. In the first example (figures 7.12 and 7.13) restoration is performed on two closely spaced noise transients. Detection of high amplitude impulsive input regions was performed using the inverse filtering detector with threshold set very high so as to avoid detecting standard clicks as noise transients. In the second example (figures 7.14, 7.15 and 7.16) a difficult section of data is taken in which there are many overlapping noise pulses of different amplitudes and shapes. This example would thus be particularly problematic for the templating method. Detection of impulsive sections was performed as for the previous example and restoration is visually very effective.

A number of passages of degraded audio material have been processed using this technique. Results have been good and usually more effective than the templating method, which requires some additional high-pass filtering to remove the last traces of degradation.

## 7.4 Kalman filter implementation

As for the single AR process models, it is possible to write the double AR process model in state-space form. This means that the separation procedure can be efficiently implemented using the Kalman filter (see section 4.4), which is an important consideration since the size of matrices to be inverted in the direct implementation can run into many hundreds. Details of such an implementation can be found in [85].

## 7.5 Conclusion

The separation algorithms developed in this chapter have been found to be a significant improvement over the templating method for removal of low frequency noise transients, both in terms of automation and the quality of processed results. Any further work needs to address the issue of detection for low frequency noise transients since it is still a difficult problem to distinguish between standard clicks and low frequency transients with low energy relative to the signal. In our current investigations we are looking

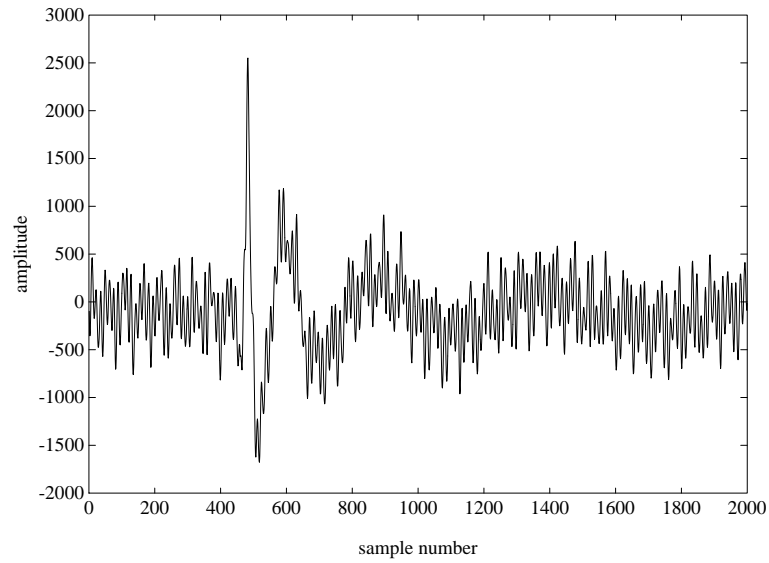


FIGURE 7.4. Audio Waveform synthetically corrupted by low frequency transient

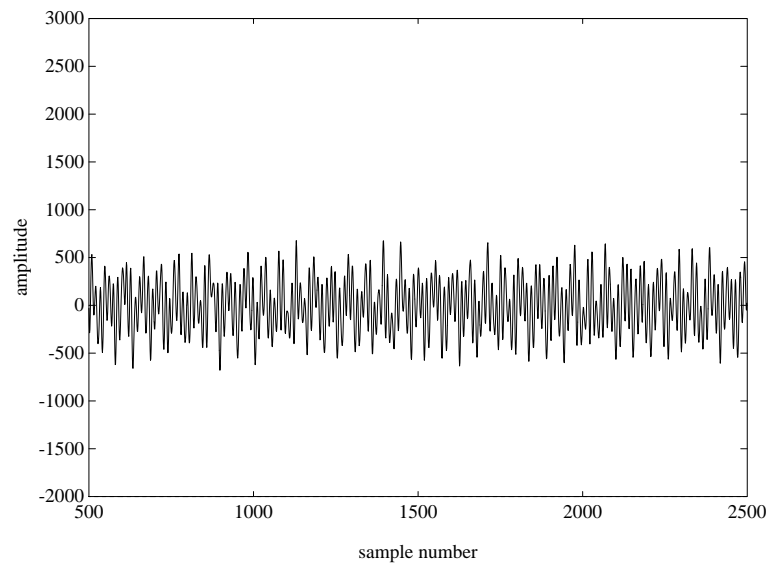


FIGURE 7.5. Restored audio data

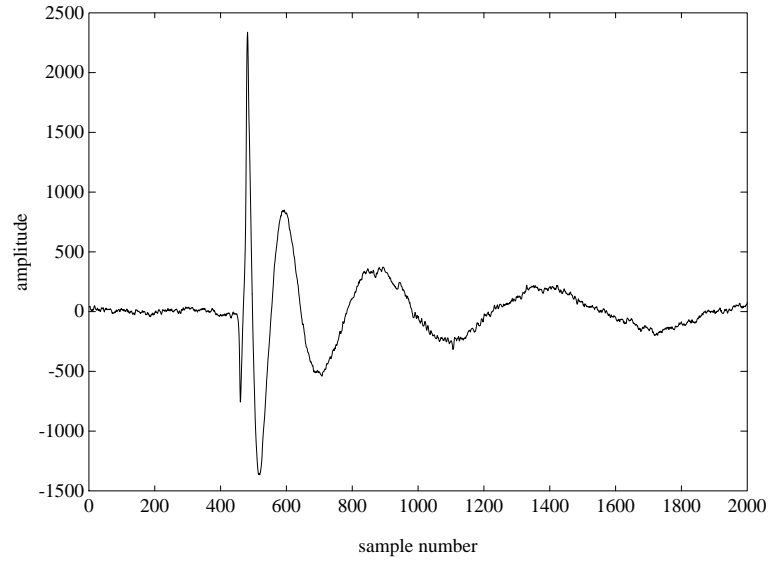


FIGURE 7.6. Genuine noise transient added to audio signal

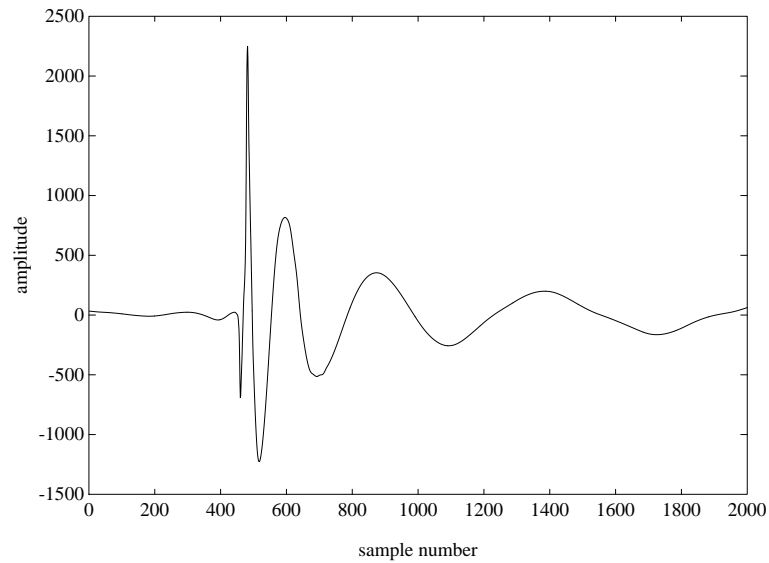


FIGURE 7.7. Noise transient estimated by restoration procedure

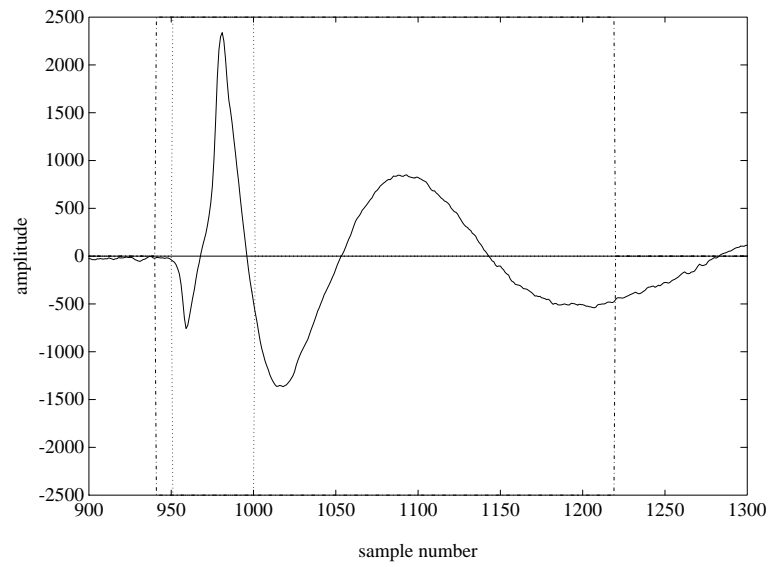


FIGURE 7.8. Genuine noise transient showing areas of signal separation and interpolation

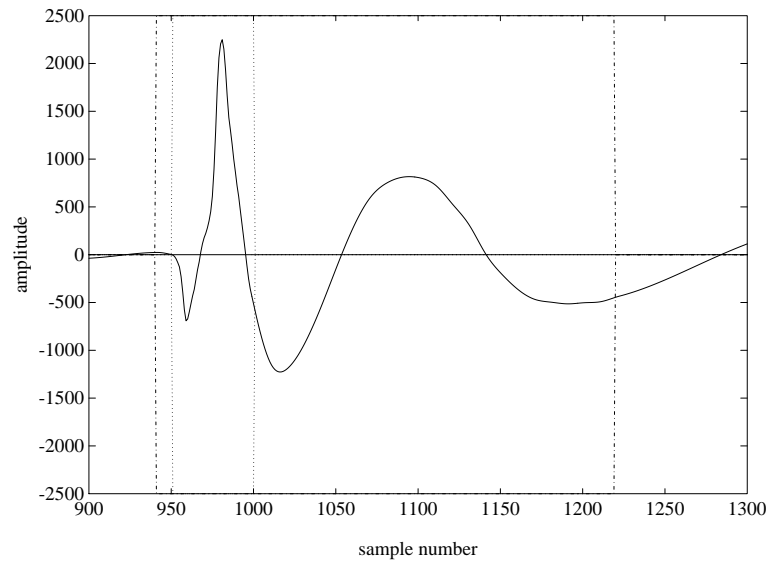


FIGURE 7.9. Noise transient estimated by restoration procedure



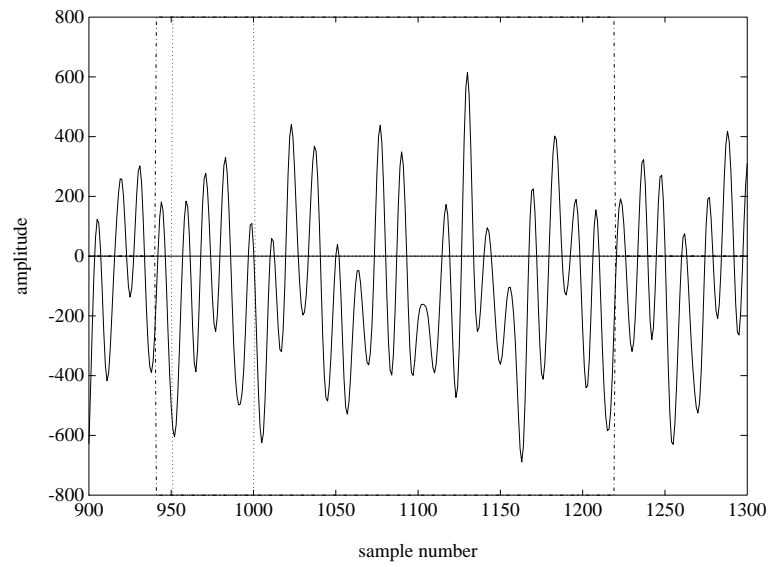


FIGURE 7.10. Original audio signal showing areas of signal separation and interpolation

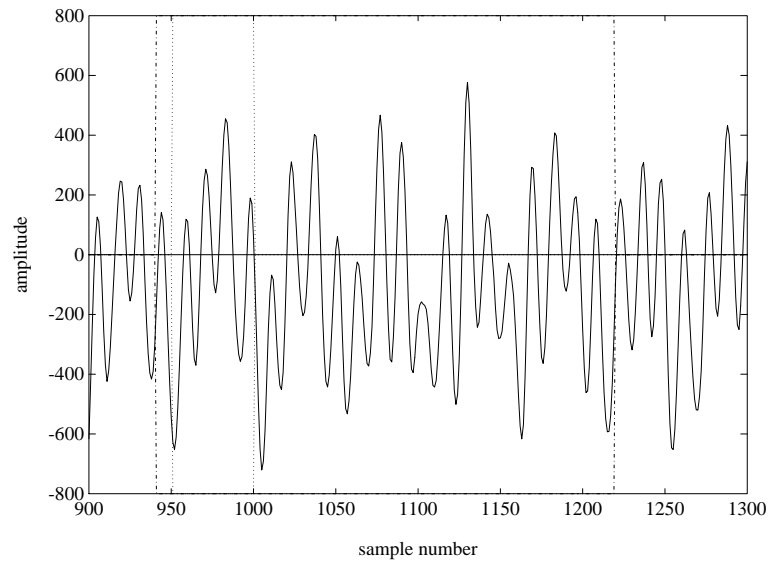


FIGURE 7.11. Signal estimated by restoration procedure

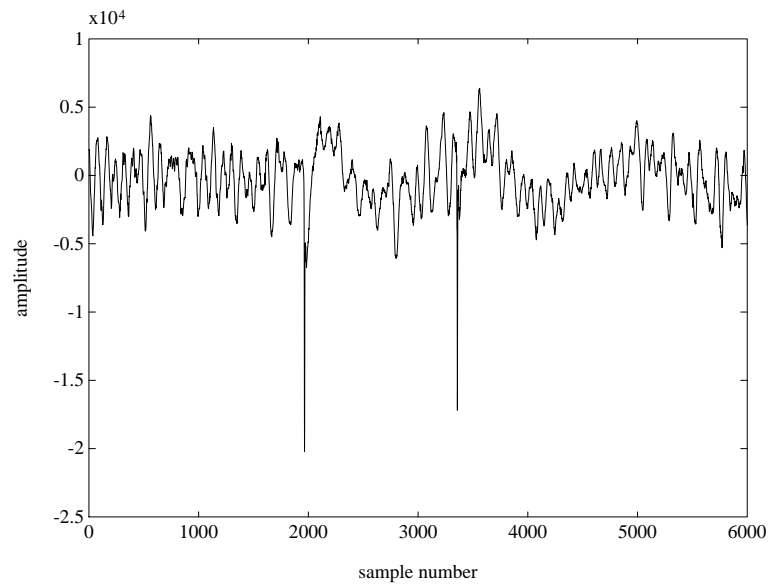


FIGURE 7.12. Degraded audio signal with two closely spaced noise transients

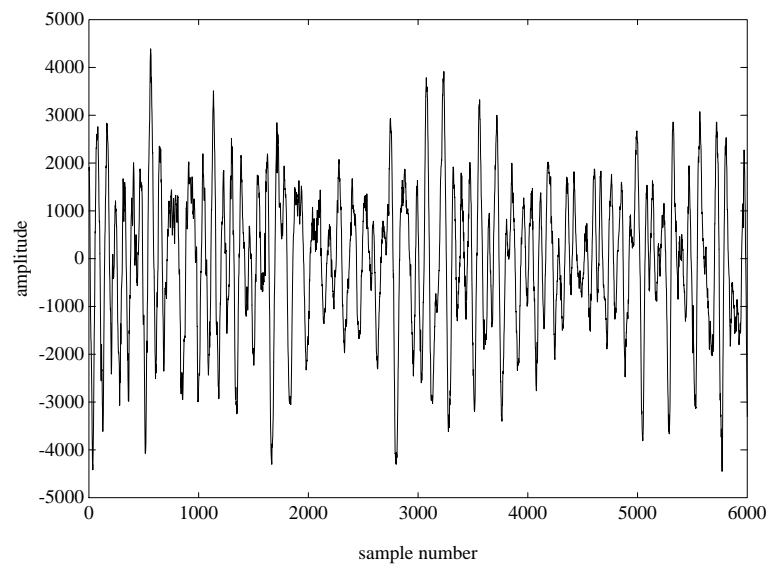


FIGURE 7.13. Restored audio signal for figure 7.12 (different scale)

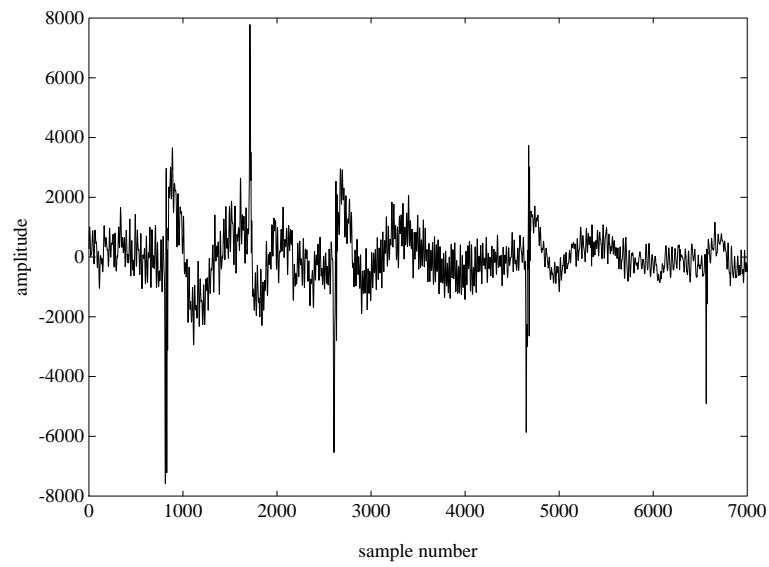


FIGURE 7.14. Degraded audio signal with many closely spaced noise transients

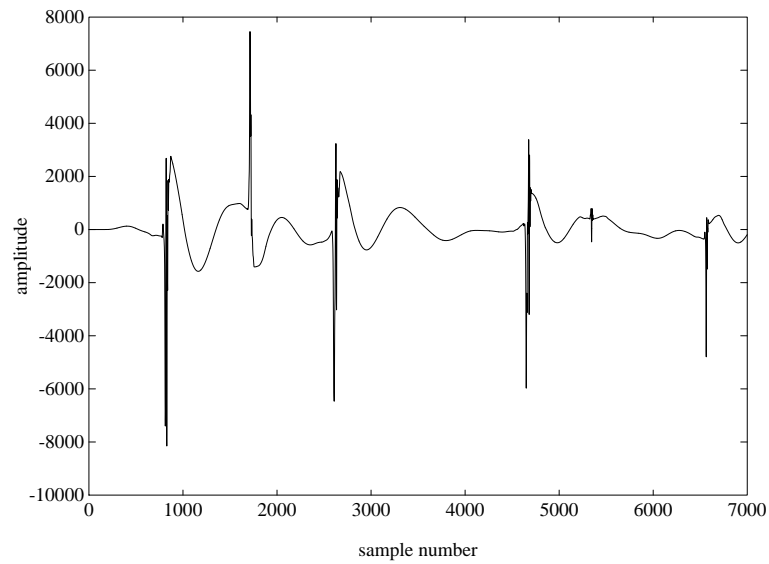


FIGURE 7.15. Estimated noise transients for figure 7.14

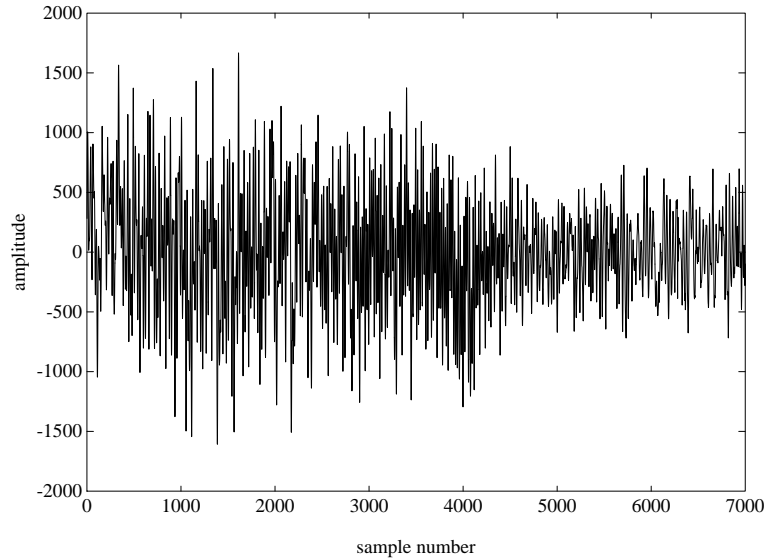


FIGURE 7.16. Restored audio signal for figure 7.14 (different scale)

at models which more accurately reflect the mechanics of the gramophone turntable system, which should lead to improved detection and restoration performance.

It is thought that the methods proposed in this chapter may find application in more general correlated transient detection and restoration. Another example from the audio field is that of high frequency resonances caused by poor choice of or faulty playback stylus. This defect has many features in common with that of low frequency transients although it has only rarely been a serious practical problem in our experience.

# 8

## Restoration of Pitch Variation Defects

A form of degradation which can be encountered on almost any analogue recording medium, including discs, magnetic tape, wax cylinders and optical film sound tracks, is an overall pitch variation which was not present in the original performance. This chapter addresses the problem of smooth pitch variation over long time-scales (e.g. variations at around 78rpm for many early gramophone discs), a defect often referred to as ‘wow’ and one of the most disturbing artefacts encountered in old recordings. An associated defect known as ‘flutter’ describes a pitch variation which varies more rapidly with time. This defect can in principle be restored within the same framework but difficulties are expected when performance effects such as tremolo or vibrato are present.

There are several mechanisms by which wow can occur. One cause is a variation of rotational speed of the recording medium during either recording or playback; this is often the case for tape or gramophone recordings. A further cause is eccentricity in the recording or playback process for disc and cylinder recordings, for example a hole which is not punched perfectly at the centre of a gramophone disc. Lastly it is possible for magnetic tape to become unevenly stretched during playback or storage; this too leads to pitch variation during playback. Early accounts of wow and flutter are given in [62, 10]. The fundamental principles behind our algorithms presented here have been presented in [78], while a recursive adaptation is given in [71].

In some cases it is in principle possible to make a mechanical correction for pitch defects, for example in the case of the poorly punched disc centre-hole, but such approaches are generally impractical. This chapter presents

a new signal processing approach to the detection and correction of these defects which is designed to be as general as possible in order to correct a wide range of related defects in recordings. No knowledge is assumed about the precise source of the pitch variation except its smooth variation with time. Results are presented from both synthetically degraded material and genuinely degraded sources.

## 8.1 Overview

The physical mechanism by which wow is produced is equivalent to a non-uniform warping of the time axis. If the undistorted time-domain waveform of the gramophone signal is written as  $x(t)$  and the time axis is warped by a function  $f_w(t)$  then the distorted signal is given by:

$$x_w(t) = x(f_w(t)) \quad (8.1)$$

For example, consider a gramophone turntable with nominal playback angular speed  $\omega_0$ . Suppose there is a periodic speed fluctuation in the turntable motor which leads to an actual speed of rotation during playback as:

$$\omega_w(t) = (1 + \alpha \cos(\omega_0 t)) \omega_0 \quad (8.2)$$

Since a given disc is designed to be replayed at constant angular speed  $\omega_0$  an effective time warping and resultant pitch variation will be observed on playback. The waveform recorded on the disc is  $x\left(\frac{\theta}{\omega_0}\right)$  where  $\theta$  denotes the angular position around the disc groove (the range  $[0, 2\pi)$  indicates the first revolution of the disk,  $[2\pi, 4\pi)$  the second, etc.) and the distorted output waveform when the motor rotates the disc at a speed of  $\omega_w(t)$  will be (assuming that  $\theta = 0$  at  $t = 0$ ):

$$\begin{aligned} x_w(t) &= x\left(\frac{1}{\omega_0} \int_0^t \omega_w(\tau) d\tau\right) \\ &= x\left(\int_0^t (1 + \alpha \cos(\omega_0 \tau)) d\tau\right) \\ &= x\left(t + \frac{1}{\omega_0} \alpha \sin(\omega_0 t)\right). \end{aligned} \quad (8.3)$$

In this case the time warping function is  $f_w(t) = t + \frac{1}{\omega_0} \alpha \sin(\omega_0 t)$ . Similar results hold for more general speed variations. The resultant pitch variation may be seen by consideration of a sinusoidal signal component  $x(t) = \sin(\omega_s t)$ . The distorted signal is then given by

$$\begin{aligned} x_w(t) &= x(f_w(t)) \\ &= \sin\left(\omega_s \left(t + \frac{1}{\omega_0} \alpha \sin(\omega_0 t)\right)\right), \end{aligned} \quad (8.4)$$

which is a frequency modulated sine wave with instantaneous frequency  $\omega_s(1 + \alpha \cos(\omega_0 t))$ . This will be perceived as a tone with time-varying pitch provided that  $\omega_s \gg \omega_0$ . Fig. 8.1 illustrates this effect. When a number of sinusoidal components of different frequencies are present simultaneously they will all be frequency modulated in such a way that there is a perceived overall variation of pitch with time.

If the time warping function  $f_w()$  is known and invertible it is possible to regenerate the undistorted waveform  $x(t)$  as

$$x(t) = x_w(f_w^{-1}(t)) \quad (8.5)$$

A wow restoration system is thus primarily concerned with estimation of the time warping function or equivalently the pitch variation function  $p_w(t) = \frac{d(f_w(t))}{dt}$ , from which the original signal can be reconstructed.

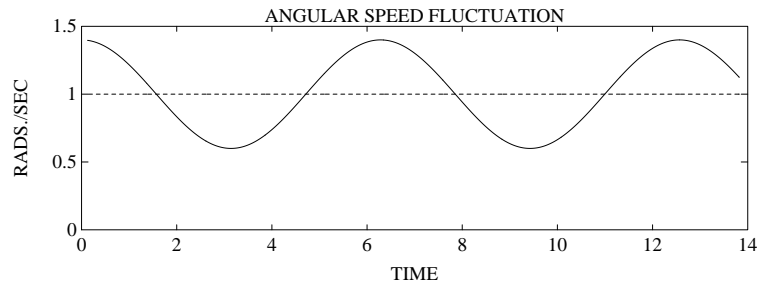
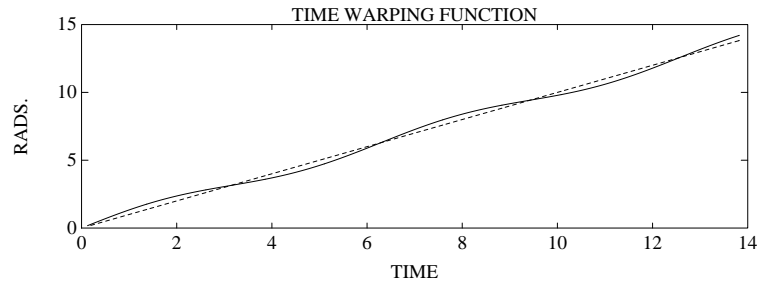
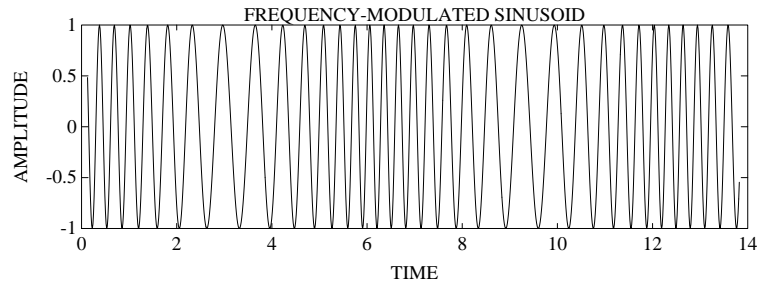
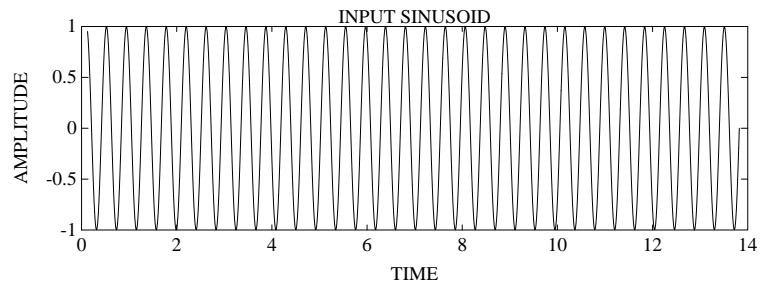
We adopt here a discrete time approach in which the values of pitch variation function are estimated from the wow-degraded data. We denote by  $\mathbf{p} = [p_1 \dots p_N]^T$  the pitch variation vector corresponding to a vector of corrupted data points  $\mathbf{x}_w$ . If it is assumed that  $\mathbf{p}$  is drawn from some random process which has prior distribution  $p(\mathbf{p})$  then Bayes' theorem gives:

$$p(\mathbf{p} \mid \mathbf{x}_w) \propto p(\mathbf{x}_w \mid \mathbf{p}) p(\mathbf{p}) \quad (8.6)$$

where the likelihood  $p(\mathbf{x}_w \mid \mathbf{p})$  can in principle be obtained from the prior for the undistorted data  $p(\mathbf{x})$  and the pitch variation vector. In fact we adopt a pre-processing stage in which the raw data are first transformed into a time-frequency 'map' which tracks the pitch of the principle frequency components in the data ('frequency tracking'). This simplifies the calculation of the likelihood and allows a practical implementation of Bayes' theorem to the transformed data. The idea behind this is the observation that most musical signals are made up of combinations of steady tones each comprising a fundamental pitch and overtones. The assumption is made that pitch variations which are common to all tones present may be attributed to the wow degradation, while other variations are genuine pitch variations in the musical performance. The approach can of course fail during non-tonal ('unvoiced') passages or if note 'slides' dominate the spectrum, but this has not usually been a problem.

Once frequency tracks have been generated they are combined using the Bayesian algorithm into a single pitch variation vector  $\mathbf{p}$ . The final restoration stage of equation (8.5) is then approximated by a digital re-sampling operation on the raw data which is effectively a time-varying sample rate converter.

The complete wow identification and restoration scheme is summarised in figure 8.2.

(a) Periodic variation of  $w_w(t)$ (b) Corresponding time-warping function  $f_w(t)$ 

(c) Single sinusoidal component

FIGURE 8.1. Frequency modulation of sinusoid due to motor speed variation.



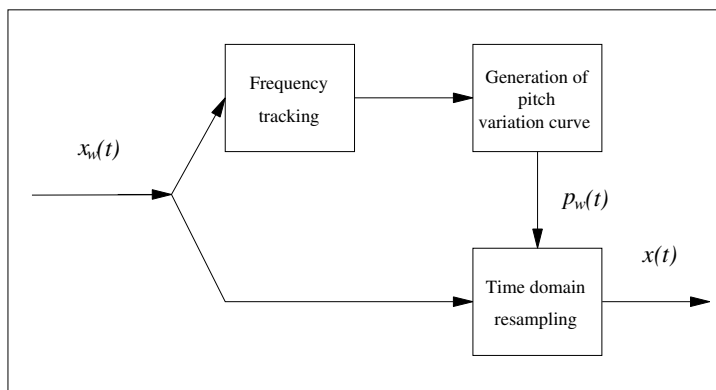


FIGURE 8.2. Outline of wow restoration system.

## 8.2 Frequency tracking

The function of the frequency tracking stage is to identify as many tonal frequency components as possible from the data and to make estimates of the way their frequencies vary with time. In principle any suitable time-varying spectral estimator may be used for this purpose (see e.g. [155, 39]). For the purpose of wow estimation it has been found adequate to employ the discrete Fourier Transform (DFT) for the frequency estimation task, although more sophisticated methods could easily be incorporated into the same framework. The window length is chosen to be short enough that frequency components within a single block are nearly constant, typically 2048 data points at 44.1kHz sampling rates, and a windowing function with suitable frequency domain characteristics such as the Hamming window is employed. Analysis of the peaks in the DFT magnitude spectrum then leads to estimates for the instantaneous frequency of tonal components in each data block. In order to obtain a set of contiguous frequency ‘tracks’ from block to block, a peak matching algorithm is used. This is closely related to the methods employed for sinusoidal coding of speech and audio, see [126, 158] and references therein.

In our scheme the position and amplitude of maxima (peaks) are extracted from the DFT magnitude spectrum for each new data block. The frequency at which the peak occurs can be refined to higher accuracy by evaluation of the DTFT on a finer grid of frequencies than the raw DFT, and a binary search procedure is used to achieve this. Peaks below a chosen threshold are deemed noise and discarded. The remaining peaks are split into two categories: those which fit closely in amplitude and frequency with an existing frequency ‘track’ from the previous block (these are added to the end of the existing track) and those which do not match with an existing track (these are placed at the start of a new frequency track). Thus

the frequency tracks evolve with time in a consistent manner, enabling the subsequent processing to estimate a pitch variation curve.

Experimental results for frequency tracking are shown in Figs. 8.4 & 8.9 and discussed in the experimental section.

### 8.3 Generation of pitch variation curve

Once frequency tracks have been generated for a section of music, the pitch variation curve can be estimated. For the  $n$ th block of data there will be  $R_n$  frequency estimates corresponding to the  $R_n$  tonal components which were being tracked at block  $n$ . The  $i$ th tonal component has a nominal (unknown) centre frequency  $F_0^i$  which is assumed to remain fixed over the period of interest in the uncorrupted music data, and a measured frequency  $F_n^i$ . Variations in  $F_n^i$  are attributed to the pitch variation value  $P_n$  for block  $n$  and a noise component  $V_n^i$ . This noise component is composed both of inaccuracies in the frequency estimation stage and genuine ‘performance’ pitch deviations (hopefully small) in tonal components.

Two modelling forms are considered. In both cases frequency measurements  $F_n^i$  are expressed as the product of centre frequency  $F_0^i$  and the pitch variation value  $P_n$  in the presence of noise. In the first case a straightforward additive form for the noise is assumed:

$$F_n^i = F_0^i P_n + V_n^i, \quad (n = 1, \dots, N, \quad i = 1, \dots, R_n) \quad (8.7)$$

This would seem to be a reasonable model where the noise is assumed to be composed largely of additive frequency estimation errors from the frequency tracking stage. However, it was mentioned that the noise term must also account for true pitch variations in musical tones which might include performance effects such as tremolo and vibrato. These may well be multiplicative, which leads to the following perhaps more natural model:

$$F_n^i = F_0^i P_n V_n^i, \quad V_n^i > 0 \quad (8.8)$$

This is equivalent to making logarithmic frequency measurements of the form:

$$f_n^i = f_0^i + p_n + v_n^i \quad (8.9)$$

where lower case characters denote the natural logarithm of the corresponding upper case quantities. Such a model linearises the estimation tasks for the centre frequencies and the pitch variation, a fact which will lead to significant computational benefits over the additive noise case. A further advantage of the multiplicative model is that all of the unknowns in the log-transformed domain can be treated as random variables over the whole

real axis, which means that the Gaussian assumption can be made without fear of estimating invalid parameter values (i.e. negative frequencies and pitch variation functions).

We now consider the multiplicative noise modelling assumption in more detail, obtaining the likelihood functions which will be required for the subsequent Bayesian analysis. For full details of the use of the additive noise model, see [70, 78, 71].

For the time being we treat the number of tonal components as fixed for all blocks with  $R_n = R$ , for notational simplicity. This restriction is removed in a later section.

For the log-transformed multiplicative model of (8.9) we can write in vector notation the observation equation at times  $n$  for the logarithmically transformed variables as

$$\mathbf{f}_n = \mathbf{f}_0 + p_n \mathbf{1}_R + \mathbf{v}_n, \quad (n = 1, \dots, N) \quad (8.10)$$

where  $\mathbf{1}_m$  is the column vector with  $m$  elements containing all ones, and

$$\begin{aligned} \mathbf{f}_n &= [f_n^1 f_n^2 \dots f_n^R]^T, \\ \mathbf{f}_0 &= [f_0^1 f_0^2 \dots f_0^R]^T, \\ \mathbf{v}_n &= [v_n^1 v_n^2 \dots v_n^R]^T. \end{aligned}$$

The corresponding matrix notation for all  $N$  data blocks is then:

$$\mathbf{F} = \mathbf{f}_0 \mathbf{1}_N^T + \mathbf{1}_R \mathbf{p}^T + \mathbf{V} \quad (8.11)$$

where

$$\mathbf{F} = [\mathbf{f}_1 \mathbf{f}_2 \dots \mathbf{f}_N], \quad (8.12)$$

$$\mathbf{p} = [p_1 p_2 \dots p_N]^T, \quad (8.13)$$

$$\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \dots \mathbf{v}_N] \quad (8.14)$$

If we now assume a specific noise distribution for the log-noise terms  $\{v_n^i\}$ , it is straightforward to obtain the likelihood function for the data,  $p(\mathbf{F} | \mathbf{p}, \mathbf{f}_0)$ . The simplest case analytically assumes that the noise terms are i.i.d. Gaussian random variables with variance  $\sigma_v^2$ . This is likely to be an over-simplification, especially for the ‘modelling error’ noise component which may contain decidedly non-white noise effects such as vibrato. Nevertheless, this i.i.d. Gaussian assumption has been used for all of our work to date and very successful results have been obtained. The likelihood is then given by:

$$p(\mathbf{F} | \mathbf{p}, \mathbf{f}_0) = \frac{1}{(2\pi\sigma_v^2)^{\frac{NR}{2}}} \exp\left(-\frac{1}{2\sigma_v^2} Q\right) \quad (8.15)$$

where

$$Q = \sum_{n=1}^N \sum_{i=1}^R (v_n^i)^2 \quad (8.16)$$

since the Jacobian of the transformation  $\mathbf{V} \rightarrow \mathbf{F}$  conditioned upon  $\mathbf{f}_0$  and  $\mathbf{p}$ , defined by (8.11), is unity. We now expand  $Q$  using (8.11) to give:

$$Q = \text{trace}(\mathbf{V}\mathbf{V}^T) \quad (8.17)$$

$$= \text{trace}(\mathbf{F}\mathbf{F}^T - 2\mathbf{F}(\mathbf{1}_N \mathbf{f}_0^T + \mathbf{p}\mathbf{1}_R^T) + \mathbf{f}_0 \mathbf{1}_N^T \mathbf{1}_N \mathbf{f}_0^T + \mathbf{1}_R \mathbf{p}^T \mathbf{p} \mathbf{1}_R^T + 2\mathbf{f}_0 \mathbf{1}_N^T \mathbf{p} \mathbf{1}_R^T) \quad (8.18)$$

$$= \text{trace}(\mathbf{F}\mathbf{F}^T) - 2\mathbf{f}_0^T \mathbf{F} \mathbf{1}_N - 2\mathbf{1}_R^T \mathbf{F} \mathbf{p} + N \mathbf{f}_0^T \mathbf{f}_0 + R \mathbf{p}^T \mathbf{p} + 2\mathbf{1}_N^T \mathbf{p} \mathbf{1}_R^T \mathbf{f}_0 \quad (8.19)$$

where the result  $(\text{trace}(\mathbf{u} \mathbf{v}^T) = \mathbf{v}^T \mathbf{u})$  has been used to obtain (8.19) from (8.18).

Differentiation w.r.t.  $\mathbf{f}_0$  and  $\mathbf{p}$  gives the following partial derivatives:

$$\frac{\partial Q}{\partial \mathbf{f}_0} = -2\mathbf{F} \mathbf{1}_N + 2N \mathbf{f}_0 + 2\mathbf{1}_{R \times N} \mathbf{p} \quad (8.20)$$

$$\frac{\partial Q}{\partial \mathbf{p}} = -2\mathbf{F}^T \mathbf{1}_R + 2R \mathbf{p} + 2\mathbf{1}_{N \times R} \mathbf{f}_0 \quad (8.21)$$

where  $\mathbf{1}_{N \times M}$  is the  $(N \times M)$ -matrix containing all unity elements. A simple ML estimate for the unknown parameter vectors would be obtained by setting these gradient expressions to zero. This, however, leads to a singular system of equations, and hence the ML estimate is not well-defined for the multiplicative noise assumption. For the additive modelling assumption, a ML solution does exist, but is found to cause noise-fitting in the estimated pitch variation vector. In the methods discussed below, prior regularising information is introduced which makes the estimation problem well-posed and we find that a linear joint estimate is possible for both  $\mathbf{f}_0$  and  $\mathbf{p}$  since (8.20) and (8.21) involve no cross-terms between the unknowns.

### 8.3.1 Bayesian estimator

The ML approach assumes no prior knowledge about the pitch variation curve and as such is not particularly useful since the solution vector tends to be very noisy for the additive noise assumption (see experimental section) or ill-posed for the multiplicative assumption. The Bayesian approach is introduced here as a means of *regularising* the solution vector such that noise is rejected in order to be consistent with qualitative prior information about the wow generation process. In (8.6) the posterior distribution for unknown parameters is expressed in terms of the raw input data. We now

assume that the raw data has been pre-processed to give frequency tracks  $\mathbf{F}$ . The joint posterior distribution for  $\mathbf{p}$  and  $\mathbf{f}_0$  can then be expressed as:

$$p(\mathbf{p}, \mathbf{f}_0 \mid \mathbf{F}) \propto p(\mathbf{F} \mid \mathbf{p}, \mathbf{f}_0) p(\mathbf{p}, \mathbf{f}_0) \quad (8.22)$$

which is analogous to equation (8.6) except that the frequency track data  $\mathbf{F}$  rather than the raw time domain signal is taken as the starting point for estimation. This is sub-optimal in the sense that some information will inevitably be lost in the pre-processing step. However, from the point of view of analytical tractability working with  $\mathbf{F}$  directly is highly advantageous.

Since we are concerned only with estimating  $\mathbf{p}$  it might be argued that  $\mathbf{f}_0$  should be marginalised from the *a posteriori* distribution. In fact, for the multiplicative noise model the marginal estimate for  $\mathbf{p}$  can be shown to be identical to the joint estimate since the posterior distribution is jointly Gaussian in both  $\mathbf{p}$  and  $\mathbf{f}_0$ .

A uniform (flat) prior is assumed for  $\mathbf{f}_0$  throughout since we have no particular information known about the distribution of centre frequencies which is not contained in the frequency tracks, hence we can write  $p(\mathbf{p}, \mathbf{f}_0) \propto p(\mathbf{p})$ . The prior chosen for  $\mathbf{p}$  will depend on what is known about the source mechanism for pitch defects. For example if the defect is caused by a poorly punched centre-hole in a disc recording, a sinusoidal model with period close to the period of angular rotation of the disc may be appropriate, while for more random pitch variations, caused perhaps by stretched analogue tape or hand-cranked disc and cylinder recordings, a stochastic model must be used. We place most emphasis on the latter case, since a system with the ability to identify such defects will have more general application than a system constrained to identify a deterministic form of defect such as a sinusoid. One vital piece of prior information which can be used in many cases is that pitch variations are *smooth* with time: the mechanical speed variations which cause wow very rarely change in a sudden fashion, so we don't expect a very 'rough' waveform or any sharp discontinuities.

In all cases considered here the prior on  $\mathbf{p}$  is zero-mean and Gaussian. In the general case, then, we have  $p(\mathbf{p}) = N(\mathbf{0}, \mathbf{C}_\mathbf{p})$ , the Gaussian distribution with mean vector zero and covariance matrix  $\mathbf{C}_\mathbf{p}$  (see appendix A.2):

$$p(\mathbf{p}) = \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{C}_\mathbf{p}|^{1/2}} \exp \left( -\frac{1}{2} \mathbf{p}^T \mathbf{C}_\mathbf{p}^{-1} \mathbf{p} \right) \quad (8.23)$$

The posterior distribution can be derived directly by substituting (8.15) and (8.23) into (8.22):

$$p(\mathbf{p}, \mathbf{f}_0 \mid \mathbf{F}) \propto \exp \left( -\frac{1}{2\sigma_v^2} Q_{\text{post}} \right) \quad (8.24)$$

where

$$Q_{\text{post}} = Q + \sigma_v^2 \mathbf{p}^T \mathbf{C}_\mathbf{p}^{-1} \mathbf{p} \quad (8.25)$$

and we can then obtain the partial derivatives:

$$\frac{\partial Q_{\text{post}}}{\partial \mathbf{f}_0} = \frac{\partial Q}{\partial \mathbf{f}_0} = -2\mathbf{F}^T \mathbf{1}_N + 2N \mathbf{f}_0 + 2 \mathbf{1}_{R \times N} \mathbf{p} \quad (8.26)$$

$$\begin{aligned} \frac{\partial Q_{\text{post}}}{\partial \mathbf{p}} &= \frac{\partial Q}{\partial \mathbf{p}} + 2\sigma_v^2 \mathbf{C}_p^{-1} \mathbf{p} \\ &= -2\mathbf{F}^T \mathbf{1}_R + 2R \mathbf{p} + 2 \mathbf{1}_{N \times R} \mathbf{f}_0 + 2 \frac{\sigma_v^2}{\sigma_e^2} \mathbf{C}_p^{-1} \mathbf{p} \end{aligned} \quad (8.27)$$

If (8.26) and (8.27) are equated to zero, linear estimates for  $\mathbf{f}_0$  and  $\mathbf{p}$ , in terms of each other, result. Back-substituting for  $\mathbf{f}_0$  gives the following estimate for  $\mathbf{p}$  alone:

$$\mathbf{p}^{\text{MAP}} = \left( \left( R \mathbf{I} - \frac{R}{N} \mathbf{1}_{N \times N} \right) + \sigma_v^2 \mathbf{C}_p^{-1} \right)^{-1} \left( \mathbf{I} - \frac{1}{N} \mathbf{1}_{N \times N} \right) \mathbf{F}^T \mathbf{1}_R \quad (8.28)$$

This linear form of solution is one of the principal advantages of a multiplicative noise model over an additive model for the pitch correction problem.

### 8.3.2 Prior models

Several specific forms of prior information are now considered. The first assumes that pitch curves can be modelled as an AR process. This is quite a general formulation which can incorporate most forms of pitch defect that we have encountered. For example, the poles of the AR process can be centred upon 78/60Hz and close to the unit circle in cases where the defect is known to originate from a 78rpm recording. A second form of prior information, which can be viewed as a special case of the AR model, includes the prior information already mentioned about the expected ‘smoothness’ of the pitch curves. It is usually the case that pitch variations occur in a smooth fashion with no sharp discontinuities. This intuitive form of prior information is thus a reasonable assumption. Finally, deterministic models for pitch variation are briefly considered, which might be appropriate for modelling gradual pitch shifts over a longer period of time (such as a turntable motor which gradually slowed down over a period of several seconds during the recording or playback process). We have successfully applied all three forms of prior to the correction of pitch variations in real recordings.

#### 8.3.2.1 Autoregressive (AR) model

This is in some sense the most general of the pitch variation models considered, allowing for a wide range of pitch defect mechanisms from highly random to sinusoidal, depending on the parameters of the AR model and

the driving excitation variance. Recall from chapter 4 that an AR process driven by a Gaussian white independent zero-mean noise process with variance  $\sigma_e^2$  has inverse covariance matrix  $\mathbf{C}_p^{-1} \approx \frac{\mathbf{A}^T \mathbf{A}}{\sigma_e^2}$  (see (4.48)), where  $\mathbf{A}$  is the usual matrix of AR coefficients. To obtain the MAP estimate for  $\mathbf{p}$  using an AR model prior, we simply substitute for  $\mathbf{C}_p^{-1}$  in result (8.28):

$$\mathbf{p}^{\text{MAP}} = \left( \left( R \mathbf{I} - \frac{R}{N} \mathbf{1}_{N \times N} \right) + \frac{\sigma_v^2}{\sigma_e^2} \mathbf{A}^T \mathbf{A} \right)^{-1} \left( \mathbf{I} - \frac{1}{N} \mathbf{1}_{N \times N} \right) \mathbf{F}^T \mathbf{1}_R \quad (8.29)$$

Note, however, that the estimate obtained assumes knowledge of the AR coefficients as well as the ratio  $\lambda = \sigma_v^2 / \sigma_e^2$ . These are generally unknown for a given problem and must be determined beforehand. It is possible to estimate the values for these parameters or to marginalise them from the posterior distributions. Such an approach leads to a non-linear search for the pitch variation estimate, for which the EM (expectation-maximisation) algorithm would be a suitable method; we do not investigate this here, however, and thus the parameter values are considered as part of the prior modelling information. For the general AR modelling case this might seem impractical. However, it is often possible to choose parameters for a low order model ( $P = 2$  or  $3$ ) by appropriate positioning of the poles of the AR filter to reflect any periodicity in the pitch deviations (such as a 78/60Hz effect for a 78rpm recording) and selecting a noise variance ratio which gives suitable pitch variation estimates over a number of trial processing blocks. When sufficient prior information is not available to position the AR poles, the smoothness prior of the next section may be more appropriate. In this case all parameter values except for the noise variance ratio are fixed. The noise variance ratio then expresses the expected ‘smoothness’ of pitch curves.

### 8.3.2.2 Smoothness Model

This prior is specifically designed to maximise some objective measure of ‘smoothness’ in the pitch variation curve. This is reasonable since it can usually be assumed that pitch defects vary in a smooth fashion. Such an approach is well known for the regularisation of least-squares and Bayesian solutions (see, e.g. [198, 150, 117, 103]). One suitable smoothness measure for the continuous case is the integral of order  $q$  derivatives:

$$\int_0^t \left| \frac{d^q p_w(\tau)}{d\tau^q} \right|^2 d\tau \quad (8.30)$$

where the time interval for wow estimation is  $(0, t)$ . In the discrete case the derivatives can be approximated by finite differencing of the time series: the first difference is given by  $d_n^1 = p_n - p_{n-1}$ , the second by  $d_n^2 = d_n^1 - d_{n-1}^1$ ,

etc. Written in matrix form we can form the sum squared of the  $q$ th order differences as follows:

$$Q_S = \sum_{n=q+1}^N (d_n^q)^2 = \mathbf{p}^T \mathbf{D}_q^T \mathbf{D}_q \mathbf{p} \quad (8.31)$$

where  $\mathbf{D}_q$  is the matrix which generates the vector of order  $q$  differences from a length  $N$  vector. An appropriate Gaussian prior which favours vectors  $\mathbf{p}$  which are ‘smooth’ to order  $q$  is then

$$p(\mathbf{p}) \propto \exp\left(-\frac{\alpha}{2} Q_S\right) \quad (8.32)$$

Substitution of covariance matrix  $\mathbf{C}_\mathbf{p}^{-1} = \alpha \mathbf{D}_q^T \mathbf{D}_q$  into equation (8.28) leads to the MAP estimate for the smoothness prior model. In a similar way to the AR model we can define a regularising constant  $\lambda = \sigma_v^2 \alpha$  which reflects the degree of ‘smoothness’ expected for a particular problem.

Such a prior is thus a special case of the AR model with fixed parameters and poles on the real axis at unity radius (for  $q = 2$ , the second order case, the parameters are  $\mathbf{a} = [2 \ -1]^T$ ). The smoothness prior thus eliminates the need for AR coefficient estimation, although  $\lambda$  must still be selected. As before,  $\lambda$  could be estimated or marginalised from the posterior distribution, but the same arguments apply as for the AR modelling case. In practice it is usually possible to select a single value for  $\lambda$  which gives suitable pitch variation curve estimates for the whole of a particular musical extract.

### 8.3.2.3 Deterministic prior models for the pitch variation

In some cases there will be specific prior information available about the form of the pitch variation curve. For example, the pitch variation may be known to be sinusoidal or to consist of a linear pitch shift over a time interval of many seconds. In these cases deterministic basis functions can be added to the prior pitch model, in much the same way as for the AR+basis function interpolators described in chapter 5. We do not present the details here as they can easily be obtained through a synthesis of ideas already discussed in this and earlier chapters. We do note, however, that both sinusoidal and linear pitch shift priors have been used successfully on a number of real recordings.

### 8.3.3 Discontinuities in frequency tracks

The above results apply only to frequency tracks which are contiguous over the whole time window of interest (recall that we fixed  $R_n$ , the number of tracks at time  $n$ , to be constant for all  $n$ ). In practice, however, frequency tracks will have gaps and discontinuities corresponding to note changes



in the musical material and possibly errors in the tracking algorithm. In some cases all tracks may disappear during pauses between notes. The likelihood expressions developed above can be straightforwardly modified to incorporate these discontinuities which leads to a small change in the form of the MAP estimate.

Each frequency track is now assigned a ‘birth’ and ‘death’ index  $b_i$  and  $d_i$ , obtained from the frequency tracking algorithm, such that  $b_i$  denotes the first time index within the block at which  $f_{0i}$  is present (‘active’) and  $d_i$  the last (each track is then continuously ‘active’ between these indices).

The model equation can now be re-written for the multiplicative noise case as

$$f_n^i = \begin{cases} f_0^i + p_n + v_n^i, & b_i \leq n \leq d_i \\ 0, & \text{otherwise} \end{cases}, \quad i = 1 \dots R_{max} \quad (8.33)$$

$R_{max}$  is the total number of tonal components tracked in the interval 1 to  $N$ . At block  $n$  there are  $R_n$  active tracks, and the length of the  $i$ th track is given by  $N_i = d_i - b_i + 1$ .  $Q$  (see (8.16)) is as before defined as the sum squared of all the noise terms  $\{v_n^i\}$ , and differentiation of  $Q$  yields the following modified gradient expressions (c.f. (8.20)-(8.21)):

$$\frac{\partial Q}{\partial \mathbf{f}_0} = -2\mathbf{F} \mathbf{1}_N + 2 \text{diag}(N_1, N_2, \dots, N_{R_{max}}) \mathbf{f}_0 + 2 \mathbf{M} \mathbf{p} \quad (8.34)$$

$$\frac{\partial Q}{\partial \mathbf{p}} = -2 \mathbf{F}^T \mathbf{1}_{R_{max}} + 2 \text{diag}(R_1, R_2, \dots, R_N) \mathbf{p} + 2 \mathbf{M}^T \mathbf{f}_0 \quad (8.35)$$

$[\mathbf{F}]_{pq} = f_q^p$  and  $\mathbf{M}$  is the  $(R_{max} \times N)$  matrix defined as:

$$[\mathbf{M}]_{pq} = \begin{cases} 1, & [\mathbf{F}]_{pq} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (8.36)$$

In other words elements of  $\mathbf{M}$  ‘flag’ whether a particular frequency track element in  $\mathbf{F}$  is active. This modified form for  $Q$  and its derivatives are substituted directly for (8.20) and (8.21) above to give a slightly modified form of the Bayesian estimator (8.28) which is used for pitch estimation on real data.

#### 8.3.4 Experimental results in pitch curve generation

Some examples of processing to generate pitch variation curve estimates are now presented. Two musical extracts sampled at 44.1kHz are used for the experiments.

The first example, ‘Viola’, is a simple extract of string music taken from a modern CD quality digital recording. There are no significant pitch defects on the recording so artificial pitch defects are generated to test the effectiveness of the pitch variation algorithms. The artificial pitch variation

is sinusoidal (see figure 8.3). This was applied to the uncorrupted signal using a resampling operation similar to that discussed shortly in section 8.4.

The second example, ‘Midsum’, is taken from a poor quality 78rpm disc recording of orchestral music which exhibits a large amount of ‘wow’ degradation. The only available source for the recording is a second or third generation analogue tape copy of the 78rpm master so it is not known whether the ‘wow’ originates from the disc or tape medium and whether a regular or periodic degradation can be assumed. The high broad-band noise content of this source means that this is a testing environment for the frequency tracking algorithm.

Both extracts have initially been pre-processed to generate frequency tracks which are shown in figures 8.4 and 8.9. Frequency tracking is limited to the range 100Hz-1kHz and a new set of frequency track values is generated once every 2048 input samples for a block of 4096 data samples (i.e. a 50% block overlap scheme).

#### 8.3.4.1 Trials using extract ‘Viola’ (synthetic pitch variation)

The first set of results, applied to extract ‘Viola’, are from the application of the second differencing ( $q = 2$ ) smoothness model (see section 8.3.2.2). This model has been found to perform well for a wide variety of pitch variation scenarios. If there is no particular reason to assume periodicity in the pitch variation then this model is probably the most robust choice. Figure 8.5 shows the estimated pitch curves using this model under both additive and multiplicative noise assumptions. The noise variance ratios  $\lambda$  have been chosen to give results with an appropriate degree of smoothness from several ‘pilot’ runs with different values of  $\lambda$ . The graph shows very little difference between the additive and multiplicative assumptions, which both give a good approximation to the true pitch curve. For pitch variations with peak-to-peak deviations within a few percent very little difference in performance is generally observed from applying the two noise assumptions. Thus the multiplicative assumption, with its computational advantage, is adopted for remaining trials on extract ‘Viola’.

Figure 8.6 shows estimated pitch curves under the AR model and sinusoidal model assumptions. Both of these models are suited to pitch variation curves which exhibit some periodicity, so performance is improved for this example compared with the smoothness prior. For the AR model an order 2 system was selected with poles close to the unit circle. The pole angle was selected by hand ( $\theta = 0.13\pi$  in this case). This may seem artificial but in a real-life situation the periodicity will often be known to a reasonable approximation (78rpm will be typical!) so a pole position in the correct range can easily be selected. For this example the resulting pitch curve was found to be fairly insensitive to errors in pole angle within  $\pm 30\%$ . As should be expected the sinusoidal model gives the best performance of all the models

in this artificial case since the true pitch variation curve is itself sinusoidal. Nevertheless the close agreement between the sinusoid model curve and the true pitch curve gives an indication that a reasonable formulation is being used for pitch curve estimation from the frequency tracks.

Figures 8.7 and 8.8 show the effect of varying  $\lambda$  over a wide range for the smoothness model and also for the AR model with poles at angle  $\pm 0.13\pi$  on the unit circle.  $\lambda = 1$  corresponds to very little prior influence on the pitch variation curve. This estimate identifies general trends in the pitch curve but since there is very little regularisation of the solution this estimate is very noisy. Increasing  $\lambda$  leads to increasingly smooth solutions. In the case of the smoothness prior (figure 8.8) the solution tends to zero amplitude as  $\lambda$  becomes large since its favoured solution is a straight line. Hence the solution is sensitive to choice of  $\lambda$ . The AR model solution (figure 8.10) is less sensitive to  $\lambda$  since its favoured solution is now a sinusoid at normalised frequency  $\theta$ , the pole angle. This feature is one possible benefit from using the AR model when periodic behaviour can be expected.

#### 8.3.4.2 Trials using extract ‘Midsum’ (non-synthetic pitch variation)

Figures 8.9 and 8.10 show frequency tracks and estimated pitch curves respectively for example ‘Midsum’. Pitch curves are estimated using the differential smoothness model under both additive and multiplicative noise assumptions. Once again little significant difference is observed between the two assumptions, which both give reasonable pitch curves. The appearance of these curves as well as the application of AR and sinusoid models (results not shown) indicates that the smoothness model is the best choice here since there is little periodicity evident in the frequency tracks. Listening tests performed with restorations from these pitch curves (see next section) confirm this.

## 8.4 Restoration of the signal

The generation of a pitch variation curve allows the final restoration stage to proceed. Equation (8.5) shows that, in principle, perfect reconstruction of the undegraded signal is possible in the continuous time case, provided the time warping function is known. In the case of band-limited discrete signals perfect reconstruction will also be possible by non-uniform resampling of the degraded samples. The degraded signal  $x_w(nT)$  is considered to be non-uniform samples from the undegraded signal  $x(t)$  with non-uniform sampling instants given by the time-warping function  $f_w(nT)$ , where  $1/T$  is the sample rate for  $x(t)$ . Provided the average sampling rate for the non-uniformly sampled signal  $x(f_w(nT))$  is greater than the Nyquist rate for  $x(t)$  it is then possible to reconstruct  $x(nT)$  perfectly from  $x(f_w(nT))$  (see e.g. [122]). Note however that the pitch varies very slowly relative to the

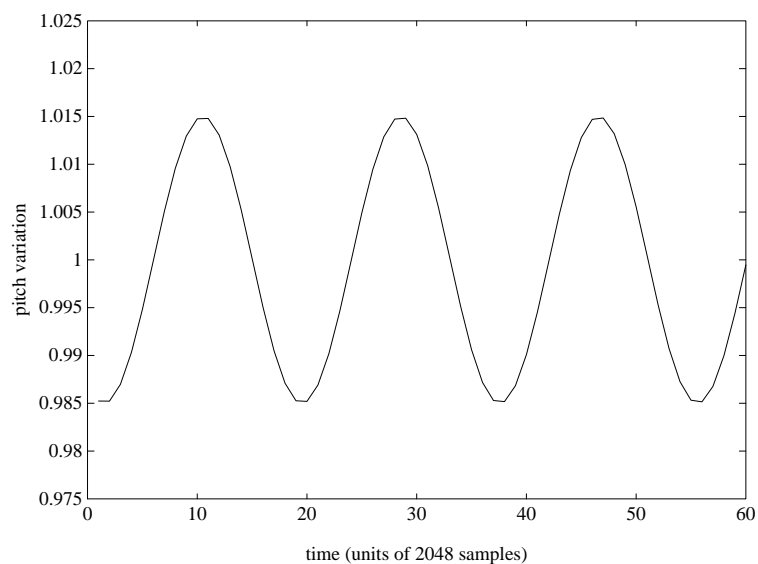


FIGURE 8.3. Synthetically generated pitch variation.

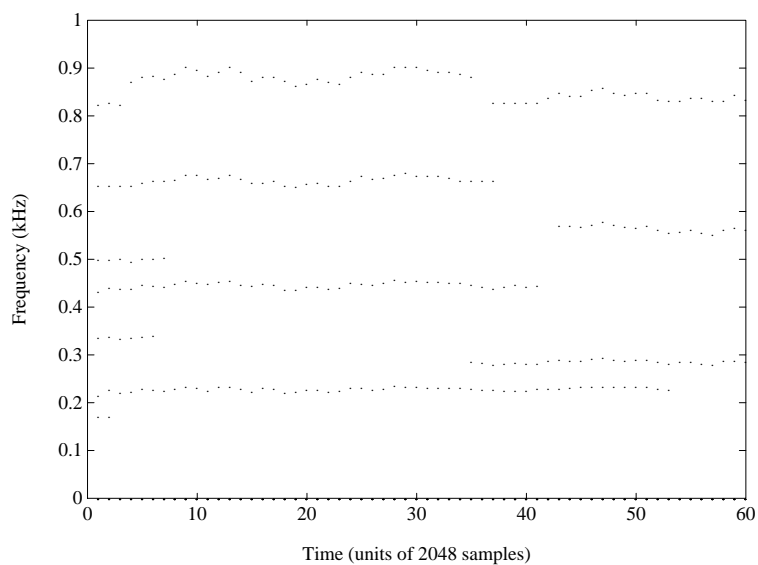


FIGURE 8.4. Frequency tracks generated for example 'Viola'.

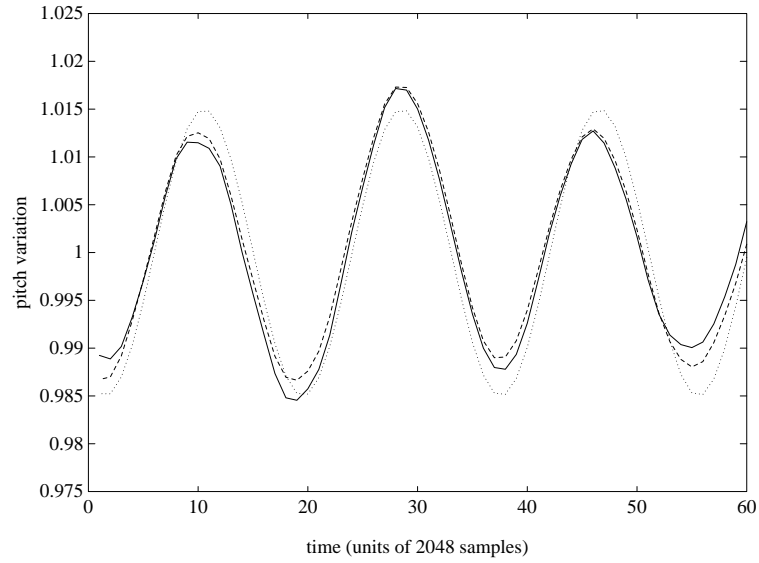


FIGURE 8.5. Pitch variation estimates for example 'Viola' using differential smoothness model: dotted line - true variation; solid line - additive model; dashed line - multiplicative model

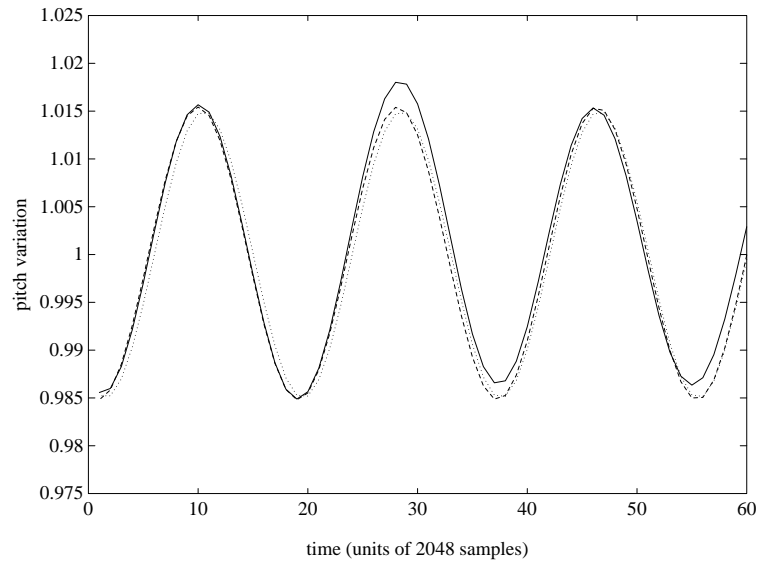


FIGURE 8.6. Pitch variation estimates for example 'Viola' using AR and sinusoid models: dotted line - true variation; solid line - AR model; dashed line - sinusoidal model

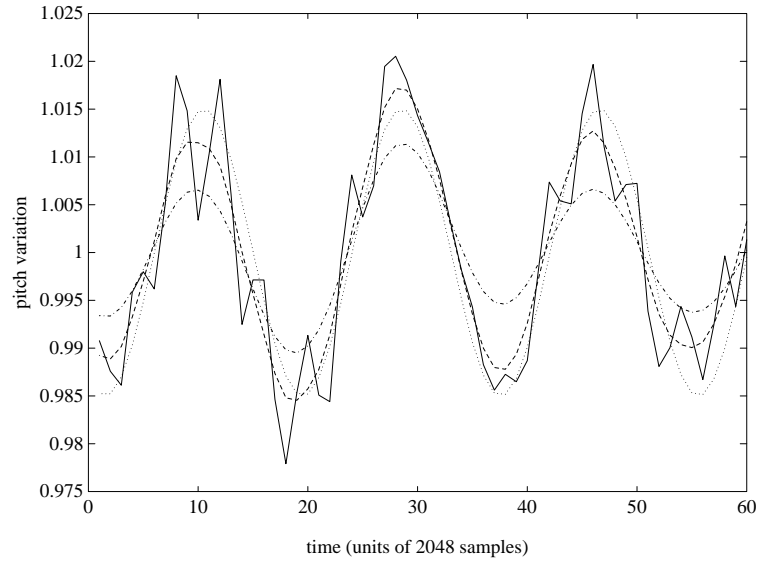


FIGURE 8.7. Pitch variation estimates for example ‘Viola’ using differential smoothness model: dotted line - true variation; solid line -  $\lambda = 1$ ; dashed line -  $\lambda = 40$ ; dot-dash line -  $\lambda = 400$

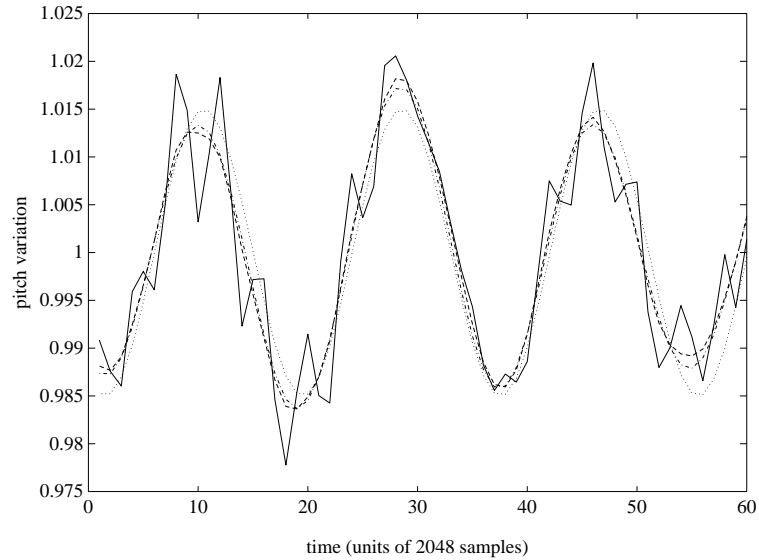


FIGURE 8.8. Pitch variation estimates for example ‘Viola’ using AR model: dotted line - true variation; solid line -  $\lambda = 1$ ; dashed line -  $\lambda = 40$ ; dot-dash line -  $\lambda = 400$

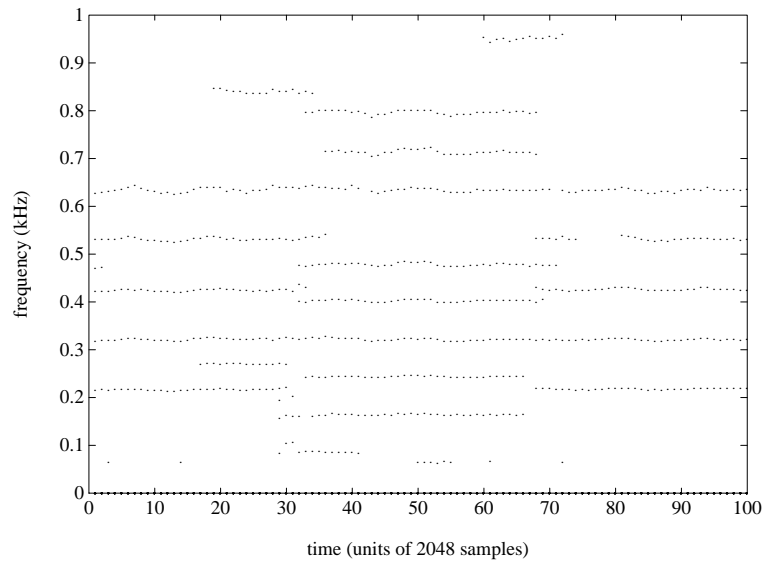


FIGURE 8.9. Frequency tracks for example 'Midsum'

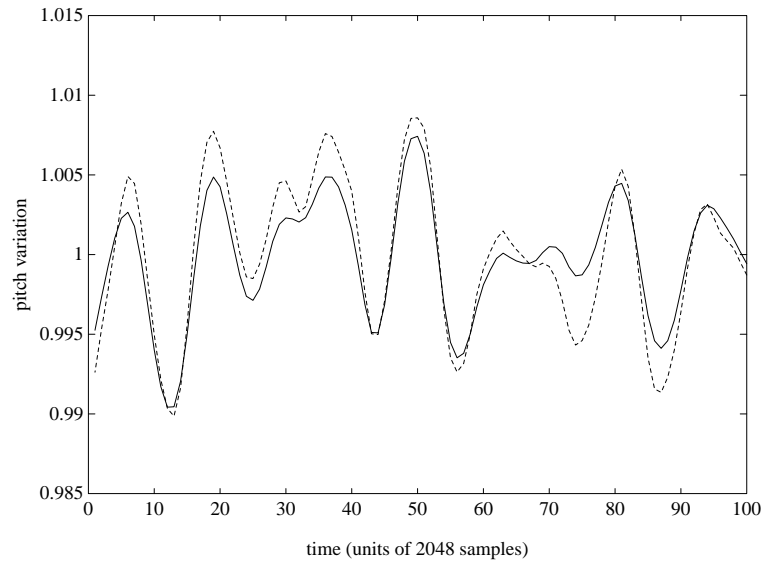


FIGURE 8.10. Pitch variation estimates for example 'Midsum' using differential smoothness model: solid line - additive model; dashed line - multiplicative model

sampling rate. Thus, at any given time instant it is possible to approximate the non-uniformly sampled input signal as a uniformly sampled signal with sample rate  $1/T' = p_w(t)/T$ . The problem is then simplified to one of *sample rate conversion* for which there are well-known techniques (see e.g. [41, 159]). Since the sample rate is slowly changing with time, a conversion method must be chosen which can easily be adjusted to arbitrary rates. A method which has been found suitable under these conditions is resampling with a truncated *sinc* function (see e.g. [41, 149]). In this method the output sample  $x(nT)$  is estimated from the closest  $(2M + 1)$  corrupted samples as

$$\hat{x}(nT) = \sum_{m=-M}^M w(mT' - \tau) \operatorname{sinc}(\alpha(mT' - \tau)) x_w((n_w - m)T') \quad (8.37)$$

where  $\alpha = \min(1, T'/T)$  ensures band-limiting to the Nyquist frequency,  $n_w$  is the closest corrupted sample to the required output sample  $n$ ,  $\tau = nT - f_w(n_w T)$  is the effective time delay and  $w(t)$  is a time domain windowing function with suitable frequency characteristics. Such a scheme can be implemented in real-time using standard digital signal processing chips.

## 8.5 Conclusion

Informal listening tests indicate that the proposed method is capable of a very high quality of restoration of gramophone recordings which would otherwise be rendered useless owing to high levels of wow. The algorithm has been tested on a wide range of pitch defects from commercial recordings and found to work robustly in many different scenarios. Improvements to the algorithms could involve a joint peak-tracking/pitch curve estimation scheme, for which some of the more sophisticated techniques discussed in chapter 12 might be applied.



# 9

## A Bayesian Approach to Click Removal

Clicks are the most common form of localised degradation encountered in audio signals. This chapter presents techniques for the removal of clicks based on an explicit model for the process by which these artefacts are generated. This work has previously been presented in [76, 79]. A full discussion and review of existing restoration methods for clicks is given in chapter 5.

Within the framework presented here, clicks are modelled as random amplitude disturbances occurring at random positions in the waveform. When a suitable form is chosen for the random processes which generate the clicks, the proposed model can accurately represent the characteristics of degradation in real audio signals. The detection and restoration procedure then becomes a problem in decision and estimation theory and we apply the Bayesian principles discussed in chapter 4. This approach allows for the incorporation of modelling information into the problem in a consistent fashion which we believe is not possible using more traditional methods. The problem is formulated such that each possible permutation of the corrupted samples in a particular section of data is represented by a separate state in a classification scheme. Bayes decision theory is then applied to identify which of these possible states is ‘most probable’. This state is chosen as the detection estimate.

Since this work is essentially concerned with the identification of aberrant observations in a time series, there are many parallels with work which has been done in the field of model-based outlier treatment in statistical data; see for example [22, 88, 13, 1, 2] for some texts of particular relevance.

This chapter develops the fundamental theory for detection using our model of click-type discontinuities. We then consider the case of AR-modelled data and particular forms of the click generating process. Classification results are derived which form the basis of both a block-based detector and a sequential detection scheme derived in the next chapter. Extensions to the basic theory are presented which consider marginalised detectors, the case of ‘noisy data’ and multi-channel detection. A special case of the detector is then shown to be equivalent to the simple detectors proposed by Vaseghi and Rayner [190, 187]. Experimental results from the methods are presented in chapter 11.

## 9.1 Modelling framework for click-type degradations

Consider a sampled data sequence  $\{x_m\}$  which is corrupted by a random, additive noise process to give a corrupted data sequence  $\{y_m\}$ . The noise samples are generated by two processes. The first is a binary (1/0) *noise generating process*,  $\{i_m\}$ , which controls a ‘switch’. The switch is connected only when  $i_m = 1$ , allowing a second noise process  $\{n_m\}$ , the *noise amplitude process*, to be added to the data signal  $x_m$ .  $y_m$  is thus expressed as:

$$y_m = x_m + i_m n_m \quad (9.1)$$

The noise source is thus a switched additive noise process which corrupts  $x_m$  only when  $i_m = 1$ . Note that the precise grouping of corrupted samples will depend on the statistics of  $\{i_m\}$ . For example, if successive values of  $i_m$  are modelled by a Markov chain (see e.g. [174]), then the transition probabilities of the Markov chain will describe some degree of ‘cohesion’ between values of  $i_m$ , i.e. the corrupting noise will tend to occur in groups or ‘bursts’. We now see how such a model can be used to represent click degradation in audio signals since, as we have observed before, clicks can be viewed as groups of corrupted samples random in length and position. The amplitude of these clicks is assumed to be generated by some random process  $\{n_m\}$  whose samples need not in general be i.i.d. and might also depend on the noise generating process  $\{i_m\}$ .

Under this model the tasks of noise detection and restoration can be defined as follows: *detection* identifies the noise switching values  $\{i_m\}$  from the observed data  $\{y_m\}$ , while *restoration* attempts to regenerate the input data  $\{x_m\}$  given observations  $\{y_m\}$ .

Note that a more realistic formulation might generalise the noise process to switch between a high amplitude ‘impulsive’ disturbance  $\{n_m^1\}$  and a low level background noise process  $\{n_m^0\}$ . This then represents a system with a continuous low level disturbance (such as broad band background

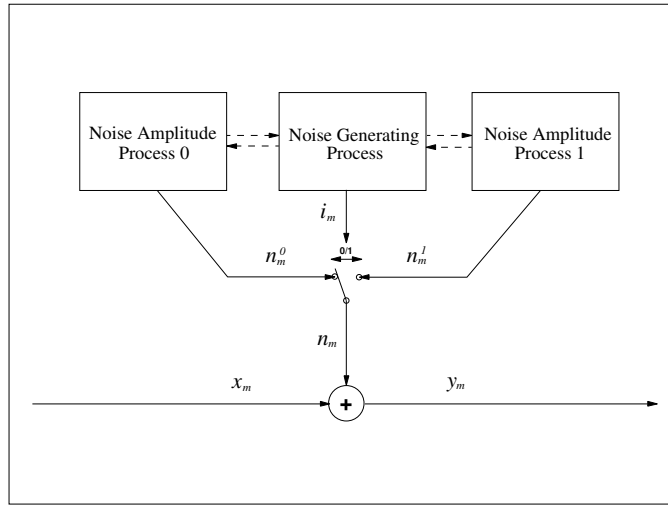


FIGURE 9.1. Schematic form for ‘noisy data’ model

noise) in addition to impulsive click-type disturbances. The corrupted waveform is modified to:

$$y_m = x_m + (1 - i_m) n_m^0 + i_m n_m^1 \quad (9.2)$$

This model will be referred to as the ‘noisy data’ case. In a later section it is shown that the optimal detector is obtained as a straightforward extension of the detector for the purely impulsive case (9.1). Figure 9.1 shows a schematic form for the noisy data model. The standard click model of (9.1) is obtained by setting  $n_m^0 = 0$ .

## 9.2 Bayesian detection

The detection of clicks is considered first and the restoration procedure is found to follow as a direct consequence. For a given vector of  $N$  observed data samples  $\mathbf{y} = [y_1 \dots y_m \dots y_N]^T$  we may write for the case of additive noise

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \quad (9.3)$$

where  $\mathbf{x}$  is the corresponding data signal and  $\mathbf{n}$  is the random additive switched noise vector containing elements  $i_m n_m$  (which can of course be zero for some or all of its elements). Define a corresponding vector  $\mathbf{i}$  with binary elements containing the corresponding samples of  $\{i_m\}$ . Note that complete knowledge of  $\mathbf{i}$  constitutes perfect detection of the corrupted samples within vector  $\mathbf{y}$ . The discrete  $N$ -vector  $\mathbf{i}$  can take on any one of  $2^N$  possible values and each value is considered to be a state within a classification framework. For the detection procedure we must estimate the true detection state (represented by  $\mathbf{i}$ ) from the observed data sequence  $\mathbf{y}$ .

Within a Bayesian classification scheme (see section 4.1.4) we can estimate the detection ‘state’  $\mathbf{i}$  by calculating the posterior probability  $p(\mathbf{i} | \mathbf{y})$  for a postulated detection state  $\mathbf{i}$ . The major part of this calculation is determination of the state *likelihood*,  $p(\mathbf{y} | \mathbf{i})$  (see 4.18) which will be the major concern of the next few sections. A general approach is derived initially in which the underlying distributions are not specified. Subsequent sections consider data modelled as an AR process with particular noise statistics.

### 9.2.1 Posterior probability for $\mathbf{i}$

If  $\mathbf{i}$  is treated as a discrete random vector whose elements may take on values 0 and 1 in any combination, we may derive the probability of  $\mathbf{i}$ , the detection state, conditional upon the corrupted data  $\mathbf{y}$  and any other prior information available to us. This posterior probability is expressed using Bayes rule as (see 4.18):

$$p(\mathbf{i} | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{i}) p(\mathbf{i})}{p(\mathbf{y})} \quad (9.4)$$

$p(\mathbf{y} | \mathbf{i})$  is the likelihood for state  $\mathbf{i}$ , which will be considered shortly.

$p(\mathbf{i})$  is the prior probability for the discrete detection vector, which will be referred to as the *noise generator prior*. This discrete probability, defined only for elements of  $\mathbf{i}$  equal to 1 or 0, reflects any knowledge we have concerning the noise generating process  $\{i_m\}$ .  $p(\mathbf{i})$  will contain any prior information about the relative probability of various click lengths and their frequencies of occurrence. A ‘uniform’ prior assigns equal prior probability to all noise configurations. However, we know from experience that this does not represent a typical noise generation process. For example, uncorrupted samples are usually far more likely than corrupted samples (a 20:1 ratio is typical), and as we have discussed in the previous section, the corruption tends to occur in ‘bursts’ of random length. A prior which expresses this knowledge may be more successful than a uniform prior which assumes all detections  $\mathbf{i}$  are equally probable. A discussion of suitable priors on  $\mathbf{i}$  is found in the next chapter.

$p(\mathbf{y})$  is a constant for any given  $\mathbf{y}$  and serves to normalise the posterior probability  $p(\mathbf{i}|\mathbf{y})$ . It can be calculated as  $p(\mathbf{y}) = \sum_{\mathbf{i}} p(\mathbf{y} | \mathbf{i}) p(\mathbf{i})$ , where the summation is over all  $2^N$  possible detection states  $\mathbf{i}$ .

For classification using a MAP procedure (see section 4.1.4) the posterior probability  $p(\mathbf{i} | \mathbf{y})$  must in principle be evaluated for all  $2^N$  possible  $\mathbf{i}$  and the estimated state is then that which maximises the posterior probability (although in the next chapter we devise sub-optimal schemes for practical operation).

The likelihood  $p(\mathbf{y} | \mathbf{i})$  is obtained from modelling considerations of both data and noise (see later).

We now adopt a notation for partitioning matrices and vectors according to the elements of  $\mathbf{i}$  which is identical to that defined in section 5.2. In other words vectors are partitioned into elements which correspond to noisy ( $i_m = 1$ ) or clean ( $i_m = 0$ ) data. Specifically the corrupt input data  $\mathbf{y}$  will be partitioned as  $\mathbf{y}_{(i)}$  and  $\mathbf{y}_{-(i)}$ , the true data  $\mathbf{x}$  will be partitioned as  $\mathbf{x}_{(i)}$  and  $\mathbf{x}_{-(i)}$  and the autoregressive parameter matrix  $\mathbf{A}$  (see 4.3 and next section) will be partitioned as  $\mathbf{A}_{(i)}$  and  $\mathbf{A}_{-(i)}$ . Clearly  $\mathbf{y}_{-(i)} = \mathbf{x}_{-(i)}$  is the section of uncorrupted data and the corrupting noise is given by  $\mathbf{n}_{(i)} = \mathbf{y}_{(i)} - \mathbf{x}_{(i)}$ .

Assume that the PDF for the corrupting noise during bursts is known to be  $p_{\mathbf{n}_{(i)}|\mathbf{i}}(\cdot)$ . In addition assume a PDF for the uncorrupted data,  $p_{\mathbf{x}}(\mathbf{x})$ . If the noise is assumed statistically independent of the data  $\mathbf{x}$  the likelihood can be obtained as (see appendix D):

$$p(\mathbf{y} | \mathbf{i}) = \int_{\mathbf{x}_{(i)}} p_{\mathbf{n}_{(i)}|\mathbf{i}}(\mathbf{y}_{(i)} - \mathbf{x}_{(i)} | \mathbf{i}) p_{\mathbf{x}}(\mathbf{U}\mathbf{x}_{(i)} + \mathbf{K}\mathbf{y}_{-(i)}) d\mathbf{x}_{(i)} \quad (9.5)$$

where  $\mathbf{U}$  and  $\mathbf{K}$  are ‘rearrangement’ matrices such that  $\mathbf{x} = \mathbf{U}\mathbf{x}_{(i)} + \mathbf{K}\mathbf{y}_{-(i)}$ , as defined in section 5.2.1. The likelihood expression can be seen to be analogous to the evidence calculation for the model selection scheme with unknown parameters (see 4.22), except we are now integrating over unknown data values rather than unknown parameters.

Note that the ‘zero’ hypothesis which tests whether  $\mathbf{i} = \mathbf{0}$ , (i.e. no samples are corrupted), is obtained straightforwardly from equation 9.4 by considering the case of  $\mathbf{i} = \mathbf{0}$ , the zero vector. No integration is required and the resulting likelihood expression is simply

$$p(\mathbf{y} | \mathbf{i} = \mathbf{0}) = p_{\mathbf{x}}(\mathbf{y}). \quad (9.6)$$

In the next section we consider detection when the signal  $\{x_m\}$  is modelled as a Gaussian AR process.

### 9.2.2 Detection for autoregressive (AR) processes

The previous section developed a general probability expression for detection of noise bursts. We now address the specific problem of detect-

ing burst-type degradation for particular audio signal models. We consider only the likelihood calculation since the posterior probability of (9.4) is then straightforward to evaluate. Inspection of (9.5) shows that we require knowledge of two probability densities:  $p_{\mathbf{n}(\mathbf{i})|\mathbf{i}}(\cdot)$ , the density for noise samples in corrupted sections, and  $p_{\mathbf{x}}(\cdot)$ , the density for the signal samples.

### 9.2.2.1 Density function for signal, $p_{\mathbf{x}}(\cdot)$

The signal model investigated here is the autoregressive (AR) model (see section 4.3), in which data samples can be expressed as a weighted sum of  $P$  previous data samples to which a random excitation value is added:

$$x_m = \sum_{i=1}^P a_i x_{m-i} + e_m. \quad (9.7)$$

The excitation sequence  $\{e_m\}$  is modelled as zero-mean white Gaussian noise of variance  $\sigma_e^2$ , and the signal  $\{x_m\}$  is assumed stationary over the region of interest. Such a model has been found suitable for many examples of speech and music and has the appeal of being a reasonable approximation to the physical model for sound production (excitation signal applied to ‘vocal tract’ filter).

The conditional likelihood for the vector of AR samples is then given by (see section 4.3.1, result (4.49)):

$$p(\mathbf{x}_1 | \mathbf{x}_0) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}\right) \quad (9.8)$$

where the expression is implicitly conditioned on the AR parameters  $\mathbf{a}$ , the excitation variance  $\sigma_e^2$  and the total number of samples (including  $\mathbf{x}_0$ ) is  $N$ .

We then adopt the approximate likelihood result (see (4.50)):

$$p_{\mathbf{x}}(\mathbf{x}) \approx p(\mathbf{x}_1 | \mathbf{x}_0), \quad N \gg P \quad (9.9)$$

As noted in section 4.3.1, this result can easily be made exact by the incorporation of the extra term  $p(\mathbf{x}_0)$ , but we omit this here for reasons of notational simplicity. Furthermore, as also noted in section 4.3.1, results from using this approximation will be exact provided that the first  $P$  data points  $\mathbf{x}_0$  are uncorrupted. For the detection procedure studied here this simplification will imply that the first  $P$  samples of  $\mathbf{i}$  are held constant at zero indicating no corruption over these samples. In a block-based scheme we could, for example, choose to make  $\mathbf{x}_0$  the last  $P$  samples of restored data from the previous data block, so we would be reasonably confident of no corrupted samples within  $\mathbf{x}_0$ .

9.2.2.2 Density function for noise amplitudes,  $p_{\mathbf{n}_{(i)}|\mathbf{i}}(\cdot)$ 

Having considered the signal model PDF  $p_{\mathbf{x}}(\cdot)$  we now consider the *noise amplitude* density function,  $p_{\mathbf{n}_{(i)}|\mathbf{i}}$ , which is the distribution of noise amplitudes within bursts, given knowledge of the noise burst positions  $\mathbf{i}$ .

The precise form for  $p_{\mathbf{n}_{(i)}|\mathbf{i}}(\cdot)$  is not in general known, but one possible assumption is that the noise samples within bursts are mutually independent and drawn from a Gaussian zero-mean process of variance  $\sigma_n^2$ . Observation of many click-degraded audio signals indicates that such a noise prior may be reasonable in many cases. Note that if more explicit information is known about the noise PDF during bursts this may be directly incorporated. In particular, a more general Gaussian distribution is easily incorporated into the algorithm as described here. The assumption of such a zero-mean multivariate Gaussian density with covariance matrix  $\mathbf{R}_{\mathbf{n}_{(i)}}$  leads to a noise amplitude density function of the form (see appendix A):

$$p_{\mathbf{n}_{(i)}|\mathbf{i}}(\mathbf{n}_{(i)} | \mathbf{i}) = \frac{1}{(2\pi)^{l/2} |\mathbf{R}_{\mathbf{n}_{(i)}}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{n}_{(i)}^T \mathbf{R}_{\mathbf{n}_{(i)}}^{-1} \mathbf{n}_{(i)}\right) \quad (9.10)$$

where  $l = (\mathbf{i}^T \mathbf{i})$  is the number of corrupted samples indicated by detection vector  $\mathbf{i}$ .

## 9.2.2.3 Likelihood for Gaussian AR data

The noise and signal priors can now be substituted from (9.10) and (9.8) into the integral of equation (9.5) to give the required likelihood for  $\mathbf{i}$ . The resulting expression is derived in full in appendix D.1 and after some manipulation may be written in the following form:

$$p(\mathbf{y} | \mathbf{i}) = \frac{\sigma_e^l \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}{(2\pi\sigma_e^2)^{\frac{N-P}{2}} |\mathbf{R}_{\mathbf{n}_{(i)}}|^{1/2} |\Phi|^{1/2}} \quad (9.11)$$

where

$$E_{\text{MIN}} = E_0 - \boldsymbol{\theta}^T \mathbf{x}_{(i)}^{\text{MAP}} \quad (9.12)$$

$$\mathbf{x}_{(i)}^{\text{MAP}} = \Phi^{-1} \boldsymbol{\theta} \quad (9.13)$$

$$\Phi = \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} + \sigma_e^2 \mathbf{R}_{\mathbf{n}_{(i)}}^{-1} \quad (9.14)$$

$$\boldsymbol{\theta} = -\left(\mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} - \sigma_e^2 \mathbf{R}_{\mathbf{n}_{(i)}}^{-1} \mathbf{y}_{(i)}\right) \quad (9.15)$$

$$E_0 = \mathbf{y}_{-(i)}^T \mathbf{A}_{-(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} + \sigma_e^2 \mathbf{y}_{(i)}^T \mathbf{R}_{\mathbf{n}_{(i)}}^{-1} \mathbf{y}_{(i)} \quad (9.16)$$

Equations (9.6) and (9.8) lead to the following straightforward result for the likelihood for no samples being erroneous ( $\mathbf{i} = \mathbf{0}$ ):

$$p(\mathbf{y} | \mathbf{i} = \mathbf{0}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right) \quad (9.17)$$

where

$$E_{\text{MIN}} = \mathbf{y}^T \mathbf{A}^T \mathbf{A} \mathbf{y}. \quad (9.18)$$

The term  $\mathbf{x}_{(i)}^{\text{MAP}}$  (9.13) is the MAP estimate for the unknown data  $\mathbf{x}_{(i)}$  given the detection state  $\mathbf{i}$  and is calculated as a byproduct of the likelihood calculation of (9.11)-(9.16).  $\mathbf{x}_{(i)}^{\text{MAP}}$  has already been derived and discussed in the earlier click removal chapter in section 5.2.3.4. Under the given modelling assumptions for signal and noise,  $\mathbf{x}_{(i)}^{\text{MAP}}$  is clearly a desirable choice for restoration for a given detection state  $\mathbf{i}$ , since it possesses many useful properties including MMSE (see 4.1.4) for a particular  $\mathbf{i}$ . Thus we can achieve joint detection and restoration using the Bayes classification scheme by choosing the MAP detection state  $\mathbf{i}$  and then restoring over the unknown samples for state  $\mathbf{i}$  using  $\mathbf{x}_{(i)}^{\text{MAP}}$ . This is a suboptimal procedure, but one which is found to be perfectly adequate in practice.

A special case of the noise statistics requires the assumption that the corrupting noise samples  $\mathbf{n}_{(i)}$  are drawn from a zero-mean white Gaussian noise process of variance  $\sigma_n^2$ . In this case the noise autocorrelation matrix is simply  $\sigma_n^2 \mathbf{I}$  with  $\mathbf{I}$  the  $(l \times l)$  identity matrix, and thus  $\mathbf{R}_{\mathbf{n}_{(i)}}^{-1} = \frac{1}{\sigma_n^2} \mathbf{I}$  and  $|\mathbf{R}_{\mathbf{n}_{(i)}}| = \sigma_n^{2l}$ . In this case make the following modifications to (9.11)-(9.16):

$$p(\mathbf{y} | \mathbf{i}) = \frac{\mu^l \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}{(2\pi\sigma_e^2)^{\frac{N-P}{2}} |\Phi|^{1/2}} \quad (9.19)$$

and

$$E_{\text{MIN}} = E_0 - \boldsymbol{\theta}^T \mathbf{x}_{(i)}^{\text{MAP}} \quad (9.20)$$

$$\mathbf{x}_{(i)}^{\text{MAP}} = \Phi^{-1} \boldsymbol{\theta} \quad (9.21)$$

$$\Phi = \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} + \mu^2 \mathbf{I} \quad (9.22)$$

$$\boldsymbol{\theta} = -\left(\mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} - \mu^2 \mathbf{y}_{(i)}\right) \quad (9.23)$$

$$E_0 = \mathbf{y}_{-(i)}^T \mathbf{A}_{-(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} + \mu^2 \mathbf{y}_{(i)}^T \mathbf{y}_{(i)} \quad (9.24)$$

where  $\mu = \sigma_e / \sigma_n$ . This simplification is used in much of the following work and it appears to be a reasonable model for the degradation in many audio signals.

As derived in section 5.2.2 above, the standard AR-based interpolator effectively discards the observed data over the corrupted sections. We can achieve the same result within our click modelling framework by allowing the noise variance  $\sigma_n^2$  to become very large. As the ratio  $\mu = \sigma_e / \sigma_n \rightarrow 0$  the noise distribution tends to uniform and the corrupted data is ignored. Taking the limit makes the following modifications to equations (9.22)-



(9.24):

$$\mathbf{\Phi} = \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \quad (9.25)$$

$$\boldsymbol{\theta} = -\mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} \quad (9.26)$$

$$E_0 = \mathbf{y}_{-(i)}^T \mathbf{A}_{-(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} \quad (9.27)$$

and in this case  $\mathbf{x}_{(i)}^{\text{MAP}}$  is identical to the  $\mathbf{x}_{(i)}^{\text{LS}}$ , the standard AR-based interpolation.

#### 9.2.2.4 Reformulation as a probability ratio test

The detector can be reformulated as a probability ratio test: specifically, form the ratio  $\Lambda(\mathbf{i})$  of posterior probability for detection state  $\mathbf{i}$  to that for  $\mathbf{i} = \mathbf{0}$ , the ‘null’ hypothesis. Using results (9.4), (9.11) and (9.17) we obtain

$$\Lambda(\mathbf{i}) = \frac{p(\mathbf{i} | \mathbf{y})}{p(\mathbf{i} = \mathbf{0} | \mathbf{y})} = \frac{p(\mathbf{y} | \mathbf{i}) p(\mathbf{i})}{p(\mathbf{y} | \mathbf{i} = \mathbf{0}) p(\mathbf{i} = \mathbf{0})} \quad (9.28)$$

$$= \frac{p(\mathbf{i})}{p(\mathbf{i} = \mathbf{0})} \frac{\sigma_e^l \exp\left(-\frac{1}{2\sigma_e^2} \Delta E_{\text{MIN}}\right)}{|\mathbf{R}_{\mathbf{n}(i)}|^{1/2} |\mathbf{\Phi}|^{1/2}} \quad (9.29)$$

which is the ratio of likelihoods for the two states, weighted by the relative prior probabilities  $\frac{p(\mathbf{i})}{p(\mathbf{i} = \mathbf{0})}$ .  $\Delta E_{\text{MIN}}$  is the difference between  $E_{\text{MIN}}$  for detection state  $\mathbf{i}$  (9.12) and  $E_{\text{MIN}}$  for the null state  $\mathbf{i} = \mathbf{0}$  (9.18). The whole expression is perhaps more conveniently written as a log-probability ratio:

$$\begin{aligned} \log(\Lambda(\mathbf{i})) &= \log\left(\frac{p(\mathbf{i})}{p(\mathbf{i} = \mathbf{0})}\right) + \frac{1}{2} \log\left(\frac{\sigma_e^{2l}}{|\mathbf{R}_{\mathbf{n}(i)}|}\right) \\ &\quad - \frac{1}{2} \log |\mathbf{\Phi}| - \frac{1}{2\sigma_e^2} \Delta E_{\text{MIN}} \end{aligned} \quad (9.30)$$

where the only data-dependent term is  $\Delta E_{\text{MIN}}$  and the remaining terms form a constant threshold for a given state  $\mathbf{i}$  compared with the null state  $\mathbf{i} = \mathbf{0}$ . We now consider this term in some detail since it has an intuitively appealing form and will be used later in the chapter. Expansion of  $\Delta E_{\text{MIN}}$  using (9.18) and (9.12-9.16) and some manipulation gives the result

$$\Delta E_{\text{MIN}} = -(\mathbf{y}_{(i)} - \mathbf{x}_{(i)}^{\text{MAP}})^T \mathbf{\Phi} (\mathbf{y}_{(i)} - \mathbf{x}_{(i)}^{\text{MAP}}) \quad (9.31)$$

which can be seen as the error between the corrupted data  $\mathbf{y}_{(i)}$  and the MAP data estimate  $\mathbf{x}_{(i)}^{\text{MAP}}$ , ‘weighted’ by  $\mathbf{\Phi}$ . Broadly speaking, the larger the error between interpolated signal  $\mathbf{x}_{(i)}^{\text{MAP}}$  and corrupted data  $\mathbf{y}_{(i)}$ , the higher the probability that the data is corrupted.

An alternative form for  $\Delta E_{\text{MIN}}$  is obtained by expanding  $\Phi$  and  $\mathbf{x}_{(i)}^{\text{MAP}}$  using (9.13) and (9.14) to give:

$$\begin{aligned} \Delta E_{\text{MIN}} &= - \left( \mathbf{y}_{(i)} - \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right)^{-1} \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} \right)^T \\ &\quad \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right)^T \Phi^{-1} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right) \\ &\quad \left( \mathbf{y}_{(i)} - \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right)^{-1} \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} \right) \quad (9.32) \\ &= - \left( \mathbf{y}_{(i)} - \mathbf{x}_{(i)}^{\text{LS}} \right)^T \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right)^T \Phi^{-1} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} \right) \left( \mathbf{y}_{(i)} - \mathbf{x}_{(i)}^{\text{LS}} \right) \quad (9.33) \end{aligned}$$

which shows a relationship with the LSAR interpolation  $\mathbf{x}_{(i)}^{\text{LS}}$  (see (5.15)). The probability ratio form of the Bayes detector may often be a more useful form for practical use since we have eliminated the unknown scaling factor  $p(\mathbf{y})$  from the full posterior probability expression (9.4). Probability ratios are now measured relative to the zero detection, so we will have at least some idea how ‘good’ a particular state estimate is relative to the zero detection.

### 9.2.3 Selection of optimal detection state estimate $\mathbf{i}$

As discussed in section 4.1.4 the MAP state selection  $\mathbf{i}^{\text{MAP}}$  will correspond to minimum error-rate detection. In order to identify  $\mathbf{i}^{\text{MAP}}$  it will in general be necessary to evaluate the likelihood according to equations (9.11)-(9.16) for all states  $\mathbf{i}$  and substitute into the posterior probability expression of (9.4) with some assumed (or known) noise generator prior  $p(\mathbf{i})$ . The state which achieves maximum posterior probability is selected as the final state estimate and the corresponding MAP data estimate,  $\mathbf{x}_{(i)}^{\text{MAP}}$ , may be used to restore the corrupted samples. This scheme will form the basis of the experimental work in chapter 11. Methods are described there which reduce the number of likelihood evaluations required in a practical scheme.

It is worth noting, however, that the MAP state estimate may not always be the desired solution to a problem such as this. Recall from section 4.1.4 that the MAP state estimate is the minimum-risk state estimate for a loss function which weights all errors equally. In the block-based click detection framework the MAP estimator is equally disposed to make an estimate in which all samples in the block are incorrectly detected as an estimate in which all but one sample is correctly detected. Clearly the latter case is far more acceptable in click detection for audio and hence we should perhaps consider a modified loss function which better expresses the requirements of the problem. One possibility would be a loss function which increases in dependence upon the total *number* of incorrectly identified samples, given

by  $n_e = |\mathbf{i}_i - \mathbf{i}_j|^2$ , where  $\mathbf{i}_i$  is the estimated state and  $\mathbf{i}_j$  is the true state. The loss function for choosing  $\mathbf{i}_i$  when the true state is  $\mathbf{i}_j$  is then given by

$$\lambda(\alpha_i | \mathbf{i}_j) = f(n_e) \quad (9.34)$$

where  $f()$  is some non-decreasing function which expresses how the loss increases with the number of incorrectly detected samples. A further sophistication might introduce some trade-off between the number of false detections  $n_f$  and the number of missed detections  $n_u$ . These are defined such that  $n_f$  is the number of samples detected as corrupted which were in fact uncorrupted and  $n_u$  is the number of corrupted samples which go undetected. If  $n_f(\mathbf{i}_i, \mathbf{i}_j)$  is the number of false alarms when state  $\mathbf{i}_i$  is estimated under true state  $\mathbf{i}_j$ , and likewise for the missed detections  $n_u(\mathbf{i}_i, \mathbf{i}_j)$ , a modified loss function might be

$$\lambda(\alpha_i | \mathbf{i}_j) = f(\beta n_f(\mathbf{i}_i, \mathbf{i}_j) + (1 - \beta) n_u(\mathbf{i}_i, \mathbf{i}_j)) \quad (9.35)$$

where  $\beta$  represents the trade-off between missed detections and false alarms.

Loss functions which do not lead to the MAP solution, however, are likely to require the exhaustive calculation of (4.19) for all  $\mathbf{i}$ , which will be impractical. Sub-optimal approaches have been considered and are discussed in the next chapter on sequential detection, but for the block-based algorithm experimentation is limited to estimation of the MAP detection vector.

#### 9.2.4 Computational complexity for the block-based algorithm

Computational complexity is largely determined by the matrix inversion of equation (9.13). This will in general be an  $\mathcal{O}(l^3)$  operation, repeated for each state evidence evaluation (with  $l$  varied appropriately). Note, however, that if the detection vector  $\mathbf{i}$  is constrained to detect only single, contiguous noise bursts and a ‘guard zone’ of at least  $P$  samples is maintained both before and after the longest noise burst tested, the matrix will be Toeplitz, as for the standard LSAR interpolation. Solution of the equations may then be efficiently calculated using the Levinson-Durbin recursion, requiring  $\mathcal{O}(l^2)$  floating point operations (see [52, 173]).

### 9.3 Extensions to the Bayes detector

Several useful extensions to the theory developed in this chapter are now outlined.

#### 9.3.1 Marginalised distributions

All of the detection results presented so far are implicitly conditioned on modelling assumptions, which may include parameter values. For the Gaus-

sian AR data case the posterior distributions are conditional upon the values of the AR coefficients  $\mathbf{a}$ , the excitation variance  $\sigma_e^2$ , and the noise sample variance  $\sigma_n^2$  (when the noise samples are considered to be independent zero mean Gaussian variates, see (9.19)-(9.24)). If these parameters are unknown *a priori*, as will usually be the case in real processing situations, it may be desirable to marginalise. The resulting likelihood expressions will then not depend on parameter estimates, which can be unreliable.

The authors are unaware of any analytic result for marginalising the AR coefficients  $\mathbf{a}$  from the likelihood expression (9.19). It seems unlikely that an analytic result exists because of the complicated structure of  $E_{\text{MIN}}$  and  $|\Phi|$ , both of which depend upon the AR coefficients. Detection is thus performed using estimated AR coefficients (see chapter 11).

Under certain prior assumptions it is possible to marginalise analytically one of the noise variance terms ( $\sigma_e^2$  or  $\sigma_n^2$ ). The details are summarised in appendix E. Initial experiments indicate that the marginalised detector is more robust than the standard detector with fixed parameters, but a full investigation is not presented here.

### 9.3.2 Noisy data

Detection under the ‘noisy data’ model (9.2) is a simple extension of the work in this chapter. It is thought that use of this model may give some robustness to ‘missed’ detections, since some small amount of smoothing will be applied in restoration even when no clicks are detected. Recall that corruption is modelled as switching between two noise processes  $\{n_m^0\}$  and  $\{n_m^1\}$ , controlled by the binary switching process  $\{i_m\}$ . Making the Gaussian i.i.d. assumption for noise amplitudes, the correlation matrix for the noise samples  $\mathbf{R}_{\mathbf{n}|\mathbf{i}}$  is equal to a diagonal matrix  $\Lambda$ , whose diagonal elements  $\lambda_j$  are given by

$$\lambda_j = \sigma_0^2 + i_j(\sigma_1^2 - \sigma_0^2) \quad (9.36)$$

where  $\sigma_0^2$  and  $\sigma_1^2$  are the variances of the white Gaussian noise processes  $\{n_t^0\}$  and  $\{n_t^1\}$ , respectively. Following a simplified version of the derivation given in appendices D and D.1 the state likelihood is given by:

$$p(\mathbf{y} | \mathbf{i}) = \frac{\sigma_e^P \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}{(2\pi)^{\frac{N-P}{2}} |\Lambda|^{1/2} |\Phi|^{1/2}} \quad (9.37)$$

and

$$E_{\text{MIN}} = E_0 - \boldsymbol{\theta}^T \mathbf{x}^{\text{MAP}} \quad (9.38)$$

$$\mathbf{x}^{\text{MAP}} = \Phi^{-1} \boldsymbol{\theta} \quad (9.39)$$

$$\Phi = \mathbf{A}^T \mathbf{A} + \sigma_e^2 \Lambda^{-1} \quad (9.40)$$

$$\boldsymbol{\theta} = \sigma_e^2 \Lambda^{-1} \mathbf{y} \quad (9.41)$$

$$E_0 = \sigma_e^2 \mathbf{y}^T \Lambda^{-1} \mathbf{y} \quad (9.42)$$

This procedure is clearly more computationally expensive than the pure click detection, since calculation of  $\mathbf{x}^{\text{MAP}}$  involves the inversion of an  $(N \times N)$  matrix  $\Phi$ . A further difficulty is foreseen in parameter estimation for the noisy data case. The sequential algorithms presented in the next chapter are, however, easily extended to the noisy data case with a less serious increase in computation.

### 9.3.3 Multi-channel detection

A further topic which is of interest for click removal (and audio restoration in general) is that of multi-channel sources. By this we mean that two or more sources of the same recording are available. If degradation of these sources occurred after the production process then defects are likely to be completely different. It would thus be expected that significant processing gains can be achieved through reference to all the sources. Problems in precise alignment of the sources are, however, non-trivial (see [187, 190]) for the general case. An example where two closely aligned sources are available is where a monophonic disc recording is played back using stereophonic equipment. It is often found that artefacts in the two resulting channels are largely uncorrelated. The problem of restoration of multiple aligned sources with uncorrelated clicks has been considered from a Bayesian perspective, based on modifications to the theory of this chapter. A summary of the theory is given in [70, appendix G]. The approach is based on a simple linear filtering relationship between signals in the two channels. The resulting detector is of a similar form to the single channel detector of this chapter.

### 9.3.4 Relationship to existing AR-based detectors

It is shown in [70, appendix H] that the inverse filtering and matched filtering detectors (see section 5.3.1) are equivalent to special cases of the Bayesian detectors derived in this chapter. This equivalence applies when the Bayesian detectors are constrained to detect only single corrupted samples in any given section of data. It is found that the inverse filtering detector corresponds to detection with no ‘lookahead’, i.e. the Bayes detector is testing for corruption in the last sample of any data block. With  $P$  or more samples of lookahead the matched filtering detector results. These results give some justification to the existing detectors while also highlighting the reason for their inadequacy when corruption occurs in closely spaced random bursts of many consecutive samples.

### 9.3.5 Sequential Bayes detection

We have seen that optimal detection for the Bayesian scheme of this chapter will in general require  $2^N$  expensive evaluations of a complex likelihood

function. This is highly impractical for any useful value of  $N$ . Sub-optimal methods for overcoming this problem with a block-based approach are proposed in chapter 11, but these are somewhat unsatisfactory since they rely on a good initial estimate for the detection state  $\mathbf{i}$ . In a sequential detection framework, state estimates are updated as new data samples are input. It is then possible to treat the detection procedure as a binary tree search in which each new sample can lead to two possible detection outcomes for the new data point (corrupted or uncorrupted). Paths with low posterior probabilities can be eliminated from the search, leading to a great reduction in the number of detection states visited compared with the exhaustive search. Sequential methods are derived in the next chapter and investigated experimentally in chapter 11.

## 9.4 Discussion

The Bayes restoration system has been developed in a general form. The system has been investigated for AR data with specific noise burst models thought to be realistic for many types of audio degradation. Experimental results and discussion of practical detection-state selection schemes for both artificial and real data are presented in chapter 11. There are limitations to the full Bayesian approach largely as a result of the difficulty in selecting a small subset of candidate detection vectors for a given data block (see chapter 11). This difficulty means that constraints such as one single contiguous burst of degradation per block have to be applied which reduce the generality of the method. The recursive algorithms developed in the next chapter are aimed at overcoming some of these problems and they also place the algorithm in a sequential framework which is more natural for real-time audio applications.

In addition, it has been demonstrated that the existing AR-based detection techniques described in chapter 5 are equivalent to a special case of the full Bayesian detector. As a result of this work, new detectors with a similar form to the existing simple detectors have been developed with explicit expressions for threshold values [70, appendix I].

# 10

## Bayesian Sequential Click Removal

In this chapter a sequential detection scheme is presented which is founded on a recursive update for state posterior probabilities. This recursive form is more natural for real time signal processing applications in which new data becomes available one sample at a time. However, the main motivation is to obtain a reliable search method for the optimal detection state estimate which requires few probability evaluations. The sequential algorithm achieves this by maintaining a list of ‘candidate’ detection estimates. With the input of a new data sample the posterior probability for each member of the list is updated into two new list elements. The first of these new elements corresponds to the detection estimate with the new sample uncorrupted, while the second element is for the corrupted case. If all states in the list are retained after each set of probability updates the list will grow exponentially in size. The search procedure would then amount to the exhaustive search of a binary tree. We present a reduced search in which states are deleted from the list after each iteration according to posterior probability. If we retain a sufficient number of ‘candidate’ states at each sampling instant it is hoped that we will only rarely discard a low probability state which would have developed into the MAP state estimate some samples later.

As for the sequential Bayesian classification schemes of section 4.1.5 the main task is obtaining a recursive update for the state likelihood  $p(\mathbf{i}|\mathbf{y})$ , which is derived with the aid of the matrix inversion results given in appendix B. The resulting form is related to the Kalman filtering equations, see section 4.4.

In order to obtain the posterior state probability from the state likelihood, the noise generator prior,  $p(\mathbf{i})$ , is required. In chapter 9 a Markov chain model was proposed for noise generation. Such a model fits conveniently into the sequential framework here and is discussed in more detail in this chapter.

Given updating results for the state posterior probabilities, reduced state search algorithms are developed. Criteria for retention of state estimates are proposed which result in a method related to the reduced state Viterbi algorithms [193, 6].

The methods derived in this chapter have been presented previously as [83] and patented as [69].

## 10.1 Overview of the method

We have calculated in the previous chapter a posterior state probability for a candidate detection vector  $\mathbf{i}$ :

$$p(\mathbf{i}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{i})p(\mathbf{i})$$

and (for the MAP detection criterion) we wish to maximise this posterior probability:

$$\mathbf{i}^{\text{MAP}} = \underset{\mathbf{i}}{\operatorname{argmax}}(p(\mathbf{i}|\mathbf{y}))$$

Since this is infeasible for long data vectors we devise instead a sequential procedure which allows a more efficient sub-optimal search. Assume that at time  $n$  we have a set of candidate detection vectors  $\mathcal{I}_n = \{\mathbf{i}_n^1, \mathbf{i}_n^2, \dots, \mathbf{i}_n^{M_n}\}$ , along with their posterior probabilities  $p(\mathbf{i}_n^j|\mathbf{y}_n)$ , where the subscript ‘ $n$ ’ denotes a vector with  $n$  data points, e.g.  $\mathbf{y}_n = [y_1, \dots, y_n]^T$ . These candidates and their posterior probabilities are then updated with the input of a new data point  $y_{n+1}$ . The scheme for updating from time  $n$  to  $n+1$  can be summarised as:

1. At time  $n$  we have  $\mathbf{i}_n^j$  and  $p(\mathbf{i}_n^j|\mathbf{y}_n)$ , for  $j = 1, \dots, M_n$ .
2. Input a new data point  $y_{n+1}$ .
3. For each candidate detection vector  $\mathbf{i}_n^j$ ,  $j = 1, \dots, M_n$ , generate two new candidate detection vectors of length  $n+1$  and update their posterior probabilities:

$$\begin{aligned} \mathbf{i}_{n+1}^{2j-1} &= [\mathbf{i}_n^j \ 0]^T \\ p(\mathbf{i}_{n+1}^{2j-1}|\mathbf{y}_{n+1}) &= g(p(\mathbf{i}_n^j|\mathbf{y}_n), y_{n+1}) \\ \mathbf{i}_{n+1}^{2j} &= [\mathbf{i}_n^j \ 1]^T \\ p(\mathbf{i}_{n+1}^{2j}|\mathbf{y}_{n+1}) &= g(p(\mathbf{i}_n^j|\mathbf{y}_n), y_{n+1}) \end{aligned}$$



4. Perform a ‘culling’ step to eliminate new candidates which have low posterior probability  $p(\mathbf{i}_{n+1} | \mathbf{y}_{n+1})$
5.  $n = n + 1$ , goto 1.

Details of the culling step, which maintains the set of candidates at a practicable size, are discussed later in the chapter. The computationally intensive step is in the sequential updating of posterior probabilities (step 3.), and most of the chapter is devoted to a derivation of this from first principles. At the end of the chapter it is pointed out that the recursive probability update can also be implemented using a Kalman filter.

## 10.2 Recursive update for posterior state probability

We now derive a recursive update for the posterior detection state probability,  $p(\mathbf{i} | \mathbf{y})$ , (see equation (9.4)). As before we will be concerned mainly with a recursive update for the state likelihood,  $p(\mathbf{y} | \mathbf{i})$ , since the full posterior probability is straightforwardly obtained from this term. A Gaussian AR model is assumed for the data as in the last chapter, and the corrupting noise is assumed white and Gaussian with variance  $\sigma_n^2$ . These assumptions correspond to the likelihood expression of (9.19)-(9.24).

As in section 4.1.5 a subscript  $n$  is introduced for all variables dependent on the number of samples in the processing block. Thus  $\mathbf{i}_n$  indicates a state detection vector  $\mathbf{i}$  corresponding to the first  $n$  samples of data. The likelihood for state  $\mathbf{i}_n$  is then rewritten directly from (9.19)-(9.24) as

$$p(\mathbf{y}_n | \mathbf{i}_n) = \frac{\mu^l \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}n}\right)}{(2\pi\sigma_e^2)^{\frac{n-P}{2}} |\Phi_n|^{1/2}} \quad (10.1)$$

and

$$E_{\text{MIN}n} = E_{0n} - \boldsymbol{\theta}_n^T \mathbf{x}_{(\mathbf{i})n}^{\text{MAP}} \quad (10.2)$$

$$\mathbf{x}_{(\mathbf{i})n}^{\text{MAP}} = \Phi_n^{-1} \boldsymbol{\theta}_n \quad (10.3)$$

$$\Phi_n = \mathbf{A}_{(\mathbf{i})n}^T \mathbf{A}_{(\mathbf{i})n} + \mu^2 \mathbf{I} \quad (10.4)$$

$$\boldsymbol{\theta}_n = -\left(\mathbf{A}_{(\mathbf{i})n}^T \mathbf{A}_{-(\mathbf{i})n} \mathbf{y}_{-(\mathbf{i})n} - \mu^2 \mathbf{y}_{(\mathbf{i})n}\right) \quad (10.5)$$

$$E_{0n} = \mathbf{y}_{-(\mathbf{i})n}^T \mathbf{A}_{-(\mathbf{i})n}^T \mathbf{A}_{-(\mathbf{i})n} \mathbf{y}_{-(\mathbf{i})n} + \mu^2 \mathbf{y}_{(\mathbf{i})n}^T \mathbf{y}_{(\mathbf{i})n} \quad (10.6)$$

where  $\mu = \sigma_e/\sigma_n$  as before. The recursive likelihood update we require is then written in terms of the previous evidence value and a new input sample  $y_{n+1}$  as

$$p(\mathbf{y}_{n+1} | \mathbf{i}_{n+1}) = g(p(\mathbf{y}_n | \mathbf{i}_n), y_{n+1}) \quad (10.7)$$

and as before this update can be expressed in terms of the conditional predictive distribution for the new sample  $y_{n+1}$  (see (4.32)):

$$p(\mathbf{y}_{n+1} \mid \mathbf{i}_{n+1}) = p(y_{n+1} \mid \mathbf{y}_n, \mathbf{i}_{n+1}) p(\mathbf{y}_n \mid \mathbf{i}_n) \quad (10.8)$$

since  $p(\mathbf{y}_n \mid \mathbf{i}_n) = p(\mathbf{y}_n \mid \mathbf{i}_{n+1})$ .

With the input of a new sample  $y_{n+1}$  each detection state estimate  $\mathbf{i}_n$  is extended by one new binary element  $i_{n+1}$ :

$$\mathbf{i}_{n+1} = [\mathbf{i}_n \ i_{n+1}]^T \quad (10.9)$$

Clearly for each  $\mathbf{i}_n$  there are two possibilities for the state update, corresponding to  $i_{n+1} = 0$  ( $y_{n+1}$  uncorrupted) and  $i_{n+1} = 1$  ( $y_{n+1}$  corrupted). If  $i_{n+1} = 0$  sample  $y_{n+1}$  is considered ‘known’ and then  $l$  (the number of ‘unknown’ samples) remains the same. Correspondingly the MAP estimate of the unknown data  $\mathbf{x}_{(i)(n+1)}^{\text{MAP}}$  is of the same length as its predecessor  $\mathbf{x}_{(i)n}^{\text{MAP}}$ . In this case it will be seen that the evidence update takes a similar form to that encountered in the sequential model selection scheme discussed in section 4.1.5. However, when  $i_{n+1} = 1$  the number of unknown samples increases by one. The evidence update then takes a modified form which includes a change in system ‘order’ from  $l$  to  $(l + 1)$ . The likelihood updates for both  $i_{n+1} = 0$  and  $i_{n+1} = 1$  are now derived in appendix F and summarised in the next section.

### 10.2.1 Full update scheme.

The full update is now summarised in a form which highlights terms common to both cases  $i_{n+1} = 0$  and  $i_{n+1} = 1$ . For  $i_{n+1} = 0$  the update is exactly equivalent to (F.13)-(F.20) while for  $i_{n+1} = 1$  the two stages outlined above are combined into one set of operations.

Firstly calculate the terms:

$$\mathbf{p}_n = \mathbf{P}_n \mathbf{b}_{(i)n} \quad (10.10)$$

$$\kappa_n = 1 + \mathbf{b}_{(i)n}^T \mathbf{p} \quad (10.11)$$

$$\kappa'_n = 1 + \mu^2(\kappa_n) \quad (10.12)$$

$$\Psi_n = \mathbf{p}_n \mathbf{p}_n^T \quad (10.13)$$

Now, for  $i_{n+1} = 0$ :

$$\mathbf{k}_n = \frac{\mathbf{p}_n}{\kappa_n} \quad (10.14)$$

$$d_{n+1} = y_{n+1} - \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \quad (10.15)$$

$$\alpha_{n+1} = d_{n+1} - \mathbf{x}_{(i)\mathbf{n}}^{\text{MAP}T} \mathbf{b}_{(i)n} \quad (10.16)$$

$$\mathbf{x}_{(i)(\mathbf{n}+1)}^{\text{MAP}} = \mathbf{x}_{(i)\mathbf{n}}^{\text{MAP}} + \mathbf{k}_{n+1} \alpha_{n+1} \quad (10.17)$$

$$\varepsilon_{n+1} = d_{n+1} - \mathbf{x}_{(i)(\mathbf{n}+1)}^{\text{MAP}T} \mathbf{b}_{(i)n} \quad (10.18)$$

$$\mathbf{P}_{n+1} = \mathbf{P}_n - \frac{\Psi_n}{\kappa_n} \quad (10.19)$$

$$p(\mathbf{y}_{(n+1)} | \mathbf{i}_{(n+1)}) = \frac{p(\mathbf{y}_n | \mathbf{i}_n)}{\sqrt{2\pi\sigma_e^2}} \frac{1}{\kappa_n^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \alpha_{n+1} \varepsilon_{n+1}\right) \quad (10.20)$$

and, for  $i_{n+1} = 1$ :

$$\mathbf{x}_{(i)(\mathbf{n}+1)}^{\text{MAP}} = \begin{bmatrix} \mathbf{x}_{(i)n}^{\text{MAP}} \\ y_{n+1} - \alpha_{n+1} \end{bmatrix} + \frac{\mu^2 \alpha_{n+1}}{\kappa'_n} \begin{bmatrix} \mathbf{p}_n \\ \kappa_n \end{bmatrix} \quad (10.21)$$

$$\mathbf{P}_{n+1} = \begin{bmatrix} (\mathbf{P}_n - (\mu^2 \Psi_n)/\kappa'_n) & \mathbf{p}_n/\kappa'_n \\ \mathbf{p}_n^T/\kappa'_n & \kappa_n/\kappa'_n \end{bmatrix} \quad (10.22)$$

$$p(\mathbf{y}_{(n+1)} | \mathbf{i}_{(n+1)}) = \frac{p(\mathbf{y}_n | \mathbf{i}_n)}{\sqrt{2\pi\sigma_e^2}} \frac{\mu}{\kappa'_n^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \frac{\mu^2}{\kappa'_n} \alpha_{n+1}^2\right) \quad (10.23)$$

The update for  $i_{n+1} = 1$  requires very few calculations in addition to those required for  $i_{n+1} = 0$ , the most intensive operation being the  $\mathcal{O}(l^2)$  operation  $(\mathbf{P}_n - (\mu^2 \Psi_n)/\kappa'_n)$ .

### 10.2.2 Computational complexity

We now consider the computational complexity for the full update of (10.10)-(10.23). The whole operation is  $\mathcal{O}(l^2)$  in floating point multiplications and additions, the respective totals being (ignoring single multiples and additions)  $4l^2 + 5l + n$  multiplies and  $3l^2 + 3l + n$  additions. This is a very heavy computational load, especially as the effective block length  $n$  and consequently the number of missing samples  $l$  becomes large. There are several simple ways to alleviate this computational load.

Firstly observe that  $\mathbf{b}_n$  (see F.2) contains only  $P$  non-zero terms. Hence the partitioned vectors  $\mathbf{b}_{(i)n}$  and  $\mathbf{b}_{-(i)n}$  will contain  $l_P$  and  $P - l_P$  non-zero elements, respectively, where  $(l_P \leq P \leq l)$  is the number of samples which are unknown from the most recent  $P$  samples. Careful programming will thus immediately reduce  $\mathcal{O}(l^2)$  operations such as (10.10) to a complexity of  $\mathcal{O}(l \times l_P)$ . A further implication of the zero elements in  $\mathbf{b}_n$  is that the data dependent terms  $\alpha_{n+1}$  and  $\varepsilon_{n+1}$  in the evidence updates require only the most recent  $l_P$  samples of  $\mathbf{x}_{(i)n}^{\text{MAP}}$  and  $\mathbf{x}_{(i)(n+1)}^{\text{MAP}}$  respectively, since earlier sample values are zeroed in the products  $\mathbf{b}_{(i)n}^T \mathbf{x}_{(i)n}^{\text{MAP}}$  and  $\mathbf{b}_{(i)n}^T \mathbf{x}_{(i)(n+1)}^{\text{MAP}}$  (see (10.16) and (10.18)). A direct consequence of this is that we require only to update the last  $l_P$  elements of the last  $l_P$  rows of  $\mathbf{A}_{n+1}$ . All  $\mathcal{O}(l^2)$  operations now become  $\mathcal{O}(l_P^2)$ , which can be a very significant saving over the ‘brute force’ calculation.

### 10.2.3 Kalman filter implementation of the likelihood update

We have given thus far a derivation from first principles of the sequential likelihood update. An alternative implementation of this step uses the Kalman filter and the prediction error decomposition as described in section 4.4.1. Here the ‘hyperparameter’ of the model is the detection vector up to time  $n$ ,  $\mathbf{i}_n$ , which replaces  $\theta$  in the prediction error decomposition. We write the AR model in the same state-space form as given in section 5.2.3.5, but setting  $Z = H^T$  for all times. Now to complete the model, set  $\sigma_v^2 = \sigma_n^2$  when  $i_m = 1$  and  $\sigma_v^2 = 0$  when  $i_m = 0$ . Running the Kalman filter and calculating the terms required for the prediction error decomposition now calculates the state likelihood recursively. The computations carried out are almost identical to those derived in the sequential update formulae earlier in this chapter.

### 10.2.4 Choice of noise generator prior $p(\mathbf{i})$

We have derived updates for the detection state likelihood  $p(\mathbf{x}_n | \mathbf{i}_n)$ . Referring back to (9.4) we see that the remaining term required for the full posterior update (to within a scale factor) is the noise generator prior  $p(\mathbf{i}_n)$ . As stated above one possibility is a uniform prior, in which case the likelihood can be used unmodified for detection. However, we will often have some knowledge of the noise generating process. For example in click-degraded audio the degraded samples tend to be grouped together in bursts of 1-30 samples (hence the use of the term noise ‘burst’ in the previous sections).

One possible model would be a Bernoulli model [174] in which we could include information about the ratio of corrupted to uncorrupted samples in the state probabilities. However, this does not incorporate any information about the way samples of noise cluster together, since samples are assumed independent in such a model. A better model for this might be the Markov chain [174], where the state transition probabilities allow a preference for ‘cohesion’ within bursts of noise and uncorrupted sections.

The two-state Markov chain model fits neatly into the sequential framework of this chapter, since the prior probability for  $i_{n+1}$  depends by definition on only the previous value  $i_n$  (for the first order case). We define state transition probabilities  $P_{i_n i_{n+1}}$  in terms of:

$$P_{00} = \text{Probability of remaining in state 0 (uncorrupted)} \quad (10.24)$$

$$P_{11} = \text{Probability of remaining in state 1 (corrupted)} \quad (10.25)$$

from which the state change transition probabilities follow as  $P_{01} = 1 - P_{00}$  and  $P_{10} = 1 - P_{11}$ .

The sequential update for the noise prior is then obtained as

$$p(\mathbf{i}_{n+1}) = P_{i_n i_{n+1}} p(\mathbf{i}_n) \quad (10.26)$$

which fits well with the sequential likelihood updates derived earlier. If we are not prepared to specify any degree of cohesion between noise burst samples we can set  $P_{01} = P_{11}$  and  $P_{10} = P_{00}$  to give the Bernoulli model mentioned above. In this case all that is assumed is the average ratio of corrupted samples to uncorrupted samples. Many other noise generator priors are possible and could be incorporated into the sequential framework, but we have chosen the Markov chain model for subsequent experimentation since it appears to be a realistic representation of many click type degradations.

## 10.3 Algorithms for selection of the detection vector

The previous sections derived the required recursions for the posterior probability for a candidate detection vector. A scheme must now be devised for

selecting the best detection vector in a sequential fashion. Given a set of  $M_n$  candidate detection state estimates at sample  $n$

$$\mathcal{I}_n = \{\mathbf{i}_n^1, \mathbf{i}_n^2, \dots, \mathbf{i}_n^{M_n}\} \quad (10.27)$$

for which detection probabilities  $p(\mathbf{i}_n^j | \mathbf{y})$  have been calculated, we can select the vector with maximum probability as our current detection estimate. As a new sample  $y_{n+1}$  arrives, there are two paths that each member of  $\mathcal{I}_n$  can take (corresponding to  $i_{n+1} = 0$  or  $i_{n+1} = 1$ ). The posterior probabilities for each member of  $\mathcal{I}$  can then be updated for each case according to the sequential updating strategy derived earlier to give two new detection state estimates at sample number  $(n+1)$ . We now have a set of  $2M_n$  detection vectors each with a corresponding probability, and the new set  $\mathcal{I}_{n+1}$  must now be selected from the  $2M_n$  vectors, based upon their relative probabilities. The exhaustive search procedure would involve doubling the size of  $\mathcal{I}_n$  with each increment in  $n$ , so a ‘culling’ strategy for deleting less probable state estimates must be employed after each update procedure. Such a scheme cannot guarantee the optimal (MAP) solution, but it may be possible to achieve results which are close to optimal.

For  $n$  data samples there are  $2^n$  possible detection permutations. However, the nature of a finite order AR model leads us to expect that new data samples will contain very little information about noise burst configurations many samples in the past. Hence the retention of vectors in the current set of candidate vectors  $\mathcal{I}_n$  with different noise configurations prior to  $n_0$ , where  $n \gg n_0$ , will lead to unnecessarily large  $M_n$ , the number of candidate vectors in  $\mathcal{I}_n$ . The retention of such vectors will also be detrimental to the selection of noise-burst configurations occurring more recently than sample  $n_0$ , since important detection vectors may then have to be deleted from  $\mathcal{I}_n$  in order to keep  $M_n$  within practical limits.

Hence it is found useful to make a ‘hard-decision’ for elements of  $\mathbf{i}_n$  corresponding to samples prior to  $n - d_H$ , where  $d_H$  is an empirically chosen delay. One suitable way to make this hard-decision is to delete all vectors from  $\mathcal{I}_n$  which have elements prior to sample  $n - d_H$  different from those of the most probable element in  $\mathcal{I}_n$  (i.e. the current MAP state estimate).  $d_H$  should clearly depend on the AR model order  $P$  as well as the level of accuracy required of the detector. Under such a scheme only the most recent  $d_H$  samples of the detection state estimates in  $\mathcal{I}_n$  are allowed to vary, the earlier values being fixed. Finding the state estimate with maximum probability under this constraint is then equivalent to decoding the state of a finite state machine with  $2^{d_H}$  states. Algorithms similar to the Viterbi algorithm [193, 60] are thus appropriate. However,  $d_H$  (which may well be e.g. 20-30) is likely to be too large for evaluation of the posterior probability for all  $2^{d_H}$  states. We thus apply a ‘culling’ algorithm related to the reduced forms of the Viterbi algorithm [6] in which states are deleted from  $\mathcal{I}_n$  which have log-probabilities which are not within a certain threshold below that of the most probable member of  $\mathcal{I}_n$ . This approach is flexible in that it will

keep very few candidate vectors when the detection probability function is strongly ‘peaked’ around its maximum (i.e. the detector is relatively sure of the correct solution) but in some critical areas of uncertainty it will maintain a larger ‘stack’ of possible vectors until the uncertainty is resolved with more incoming data. A further reduction procedure limits the number of elements in  $\mathcal{I}_n$  to a maximum of  $M_{max}$  by deleting the least probable elements of  $\mathcal{I}_n$ .

### 10.3.1 Alternative risk functions

The desirability of minimising some function other than the commonly-used one-zero risk function is discussed in section (9.2.3). However, as stated in that section, it will usually be necessary to calculate  $p(\mathbf{i}_n | \mathbf{y}_n)$  for all possible  $\mathbf{i}_n$ -vectors in order to achieve this goal.

In the present case we have available the set of candidate vectors  $\mathcal{I}_n$  and their associated posterior probabilities. Suppose that we were certain that  $\mathcal{I}_n$  contained the true MAP detection state. This is equivalent to assuming that the posterior probability of any detection state estimate not present in  $\mathcal{I}_n$  has zero probability, and correspondingly there is zero risk associated with not choosing any such state estimates. The total expected risk may then be rewritten as

$$R(\alpha_i | \mathbf{y}_n) \propto \sum_{\{j: \mathbf{i}_j \in \mathcal{I}_n\}} \lambda(\alpha_i | \mathbf{i}_j) P(\mathbf{i}_j | \mathbf{y}_n) \quad (10.28)$$

where we have modified the summation of (4.19) such that only elements of  $\mathcal{I}_n$  are included.

This risk function can then be evaluated for each element of  $\mathcal{I}_n$  to give a new criterion for selection of state estimate. We cannot guarantee that the correct state vector is within  $\mathcal{I}_n$ , so such a risk calculation will be sub-optimal. It is quite possible, for example, that the culling algorithm outlined above deletes a state which would later have a relatively high probability and a low expected risk. The state estimate selected from  $\mathcal{I}_n$  might then have significantly higher risk than the deleted state. Nevertheless, initial tests with a loss function of the form (9.35) indicate that some trade-off can indeed be made between false alarms and missed detections at an acceptable overall error rate using a risk function of the form (10.28), but a thorough investigation is left as future work.

## 10.4 Summary

In this chapter we have derived a sequential form for Bayesian detection of clicks. Most of the work is concerned with deriving sequential updates for the detection state likelihood which are based on matrix inversion results

similar to those used in RLS and Kalman filtering. In fact, through use of the prediction error decomposition (see section 4.4), an alternative implementation of the whole scheme would be possible using a Kalman filter. In addition we have introduced a Markov chain prior for noise generation which is used in subsequent experimental work and is designed to model the observed tendency that clicks correspond to short ‘bursts’ of corrupted samples rather than isolated impulses in the data. In the next chapter we include implementational details for the Bayes detector and perform some experimental evaluation of the algorithms presented in this and the previous chapter.



# 11

## Implementation and Experimental Results for Bayesian Detection

In this chapter we discuss implementation issues concerned with the Bayesian detection schemes derived in the last two chapters. Experimental results from several implementations of the detector are then presented which demonstrate the performance improvements attainable using the new schemes.

Firstly, the block-based approach of chapter 9 is discussed. As has been suggested earlier, the block-based scheme is awkward to implement owing to the combinatorial explosion in the number of possible detection states as the block size  $N$  increases. Sub-optimal search procedures are proposed which will usually bring the number of detection states tested within reasonable computational limits. Some experimental results are presented which show that the Bayesian approach can improve upon existing techniques. It is not proposed that these methods are applied in practice, but we present results to illustrate the workings of the Bayesian principles. The reader who is only interested in the practical algorithm can miss out this section.

Secondly, a fuller investigation of performance is made using the more flexible sequential methods of chapter 10. Algorithms for sequential selection of detection states are specified and experimental results are presented for these algorithms using both artificial and real data sequences.

## 11.1 Block-based detection

We first consider the block-based detection scheme given in chapter 9 in which the posterior probability for a particular detection state estimate is derived for a block of  $N$  corrupted data samples  $\mathbf{y}$  under Gaussian AR data assumptions (see (9.4) and (9.11)-(9.16). This section may be omitted by readers only interested in the practical implementation of the sequential scheme. In the absence of any additional information concerning the noise-burst amplitude distribution we will make the Gaussian i.i.d. assumption which leads to the evidence result of (9.19)-(9.24). In this section a ‘uniform’ prior  $P(\mathbf{i}) = 1/2^N$  is assumed for the detection state vector  $\mathbf{i}$ , i.e. for a given data block no particular state detection estimate is initially thought to be more probable than any other. Other priors based upon the Markov chain model of click degradation are considered in the section on sequential detection.

### 11.1.1 Search procedures for the MAP detection estimate

Assuming that we have estimated values for the AR system parameters it is now possible to search for the MAP detection vector based upon the posterior probability for each  $\mathbf{i}$ . We have seen that for a given data block there are  $2^N$  possibilities for the detection state, making the exhaustive posterior calculation for all  $\mathbf{i}$  prohibitive. It is thus necessary to perform a sub-optimal search which restricts the search to a few detection states which are thought most probable.

It is proposed that an initial estimate for the detection state is obtained from one of the standard detection methods outlined in section 5.3.1, such as the inverse filtering method or the matched filtering alternative. A limited search of states ‘close’ to this estimate is then made to find a local maximum in the detection probability. Perhaps the simplest procedure makes the assumption that at most one contiguous burst of data samples is corrupted in any particular data block. This will be reasonable in cases where the corruption is fairly infrequent and the block length is not excessively large.

The starting point for the search is taken as an initial estimate given by the standard techniques. Optimisation is then performed by iterated one-dimensional searches in the space of the  $\mathbf{i}$ -vector. Supposing the first corrupted sample is  $m$  and the last corrupted sample is  $n$  (the number of degraded samples is then given by  $l = n - m + 1$ ). Initial estimates  $\{n_0, m_0\}$  are obtained from the inverse filtering or matched filter method.  $m$  is initially fixed at  $m_0$  and  $n$  is varied within some fixed search region around its estimate  $n_0$ . The posterior probability for each  $\{n, m_0\}$  combination is evaluated. The value of  $n$  which gives maximum posterior probability is selected as the new estimate  $n_1$ . With  $n$  fixed at  $n_1$ ,  $m$  is now varied around  $m_0$  over a similar search region to find a new estimate  $\{n_1, m_1\}$  with max-

imum probability. This completes one iteration. The  $i$ th iteration, starting with estimate  $\{m_{i-1}, n_{i-1}\}$  and using a search region of  $\pm\Delta$  is summarised as:

1. Evaluate posterior probability for  $m = m_{i-1}$  and  $n = (n_{i-1} - \Delta) \dots (n_{i-1} + \Delta)$ .
2. Choose  $n_i$  as the value of  $n$  which maximises the posterior probability in step 1.
3. Evaluate posterior probability for  $m = (m_i - \Delta) \dots (m_i + \Delta)$  and  $n = n_i$ .
4. Choose  $m_i$  as the value of  $m$  which maximises the posterior probability in step 3.

More than one iteration may be required, depending on the extent of search region  $\Delta$  and how close the initial estimate  $\{n_0, m_0\}$  is to the MAP estimate.

An example of this simple search procedure is given in figures 11.1-11.4. In figure 11.1 a real audio signal is shown and a sequence of samples (34-39) has been corrupted by additive Gaussian noise ( $\sigma_n/\sigma_e = 4$ ). Figure 11.2 shows the magnitude output of both the matched filter and inverse filter detectors (normalised by maximum magnitude). An AR model of order 20 has been used with parameters (including  $\sigma_e$ ) estimated directly from 1000 samples of the corrupted data by the covariance method.

We see that both filters give high amplitude output in the region of degradation and so may be thresholded to give initial estimates for the corrupted section. Figure 11.3 now shows (scaled) posterior probabilities calculated for one iteration of the one-dimensional search procedures outlined above and using the same data and AR model as before. The evidence formulation of equations (9.19)-(9.24) is used for this and subsequent trials. A uniform noise generator prior was used which favours no one detection over any other. The dashed line shows step 1. above in which probabilities are evaluated for a range of  $n$  values with  $m$  fixed at  $m_0 = 31$  (an error of 3 samples). There is a clear peak in probability at  $n = 39$ , the last degraded sample. The full line shows step 3. in which  $n$  is fixed at  $n_1 = 39$  and  $m$  is varied. Again there is a strong peak at the correct  $m$  value of 34. Finally, figure 11.4 shows a mesh plot of probability for all possible  $\{m, n\}$  combinations within the range  $\{31 - 37, 36 - 42\}$  which shows a strong local maximum at the desired detection estimate  $\{34, 39\}$  (note that any combination where  $n < m$  is assigned zero probability). The one-dimensional probability plots of the previous figure are effectively just cross-sections of this 3-d plot taken parallel to the  $m$  and  $n$  axes.

The above example demonstrates typical operation of the Bayesian detection algorithm and highlights some of the advantages obtainable. In

particular it can be seen that there is some ambiguity in detection using the standard methods (see figure 11.2). It is not possible to select a threshold for this example which will correctly detect the corrupted data without leading to ‘false alarms’. The Bayesian detector on the other hand gives a clear probability maximum which requires no heuristic thresholding or other manipulations in the detection process. A suitable probability maximum is typically found within 1-2 iterations of the search procedure, involving a total of perhaps 10-20 probability evaluations. Note that false alarms in detection are automatically tested if we use the probability ratio version of the detector (see (9.28) and following). If no posterior probability ratio evaluated exceeds unity then the ‘zero’ hypothesis is selected in which no samples are detected as corrupt.

We have seen that each direct evaluation of posterior probabilities is an  $\mathcal{O}(l^2)$  operation when the constraint of one contiguous noise burst framed by at least  $P$  uncorrupted samples is made. Even using the simple search algorithms discussed above the detection procedure is considerably more computationally intensive than the standard restoration techniques given in section 5.3.1 which require, in addition to detection, a single interpolation with  $\mathcal{O}(l^2)$  complexity. The situation is not as bad as it appears, however, since the ordered nature of the search procedures outlined above lend themselves well to efficient update methods. In particular the procedure of increasing the last degraded sample number  $n$  by one or decreasing the first sample  $m$  by one can be performed in  $\mathcal{O}(l)$  operations using techniques based upon the Levinson recursion for Toeplitz systems (see [86]). Search procedures for the MAP estimate can then be  $\mathcal{O}(l_{max}^2)$ , where  $l_{max}$  is the maximum length degradation tested. Computation is then comparable to the standard methods.

The methods discussed above are somewhat limiting in that we are constrained to one single contiguous degradation in a particular data block. We require in addition a buffer of at least  $P$  uncorrupted samples to frame the degradation in order to maintain a Toeplitz system leading to efficient probability evaluations. The procedure can however be generalised for practical use. The standard detection methods are used as an ‘event’ detector. An event in the signal will be one single burst of degradation which can generally be identified from the highest maxima of the detection waveform (see e.g. figure 11.2). Each event identified in this way is then framed by sufficient samples on either side to surround the whole degradation with at least  $P$  samples. The sub-block formed by this framing process is then processed using the search procedures given above to give the Bayesian detection estimate and (if required) the restored signal waveform. This method will work well when degradations are fairly infrequent. It is then unlikely that more than one event will occur in any detection sub-block. In cases where multiple events do occur in sub-blocks the above algorithms will ignore all but the ‘current’ event and sub-optimal performance will result. Although of course more heuristics can be added to improve the

situation, this practical limitation of the block-based approach is a significant motivation for the sequential processing methods developed in the last chapter and evaluated later in this chapter.

### 11.1.2 Experimental evaluation

Some quantitative evaluation of the block-based algorithm's performance is possible if we calculate the probability of error in detection,  $P_E$ . This represents the probability that the algorithm does not pick precisely the correct detection estimate in a given block of  $N$  samples.  $P_E$  may be estimated experimentally by performing many trials at different levels of degradation and counting the number of incorrect detections  $N_E$  as a proportion of the total number of trials  $N_T$ :

$$\hat{P}_E = \frac{N_E}{N_T} \quad (11.1)$$

and for  $N_T$  sufficiently large,  $\hat{P}_E \rightarrow P_E$ .

In this set of trials data is either synthetically generated or extracted from a real audio file and a single noise burst of random length and amplitude is added at a random position to the data. The posterior probability is evaluated on a 7 by 7 grid of  $\{m, n\}$  values such that the true values of  $m$  and  $n$  lie at the centre of the grid. A correct detection is registered when the posterior probability is maximum at the centre of the grid, while the error count increases by one each time the maximum is elsewhere in the grid. This is a highly constrained search procedure which effectively assumes that the initial estimate  $\{n_0, m_0\}$  is close to the correct solution. It does however give some insight into detector performance.

The first result shown in figure 11.5 is for a synthetic order 10 AR model. Noise burst amplitudes are generated from a Gaussian distribution of the appropriate variance. The length of noise bursts is uniformly distributed between 0 and 15 samples. The full line gives the error probability under ideal conditions when  $\mu$  is known exactly. Performance becomes steadily worse from left to right which corresponds to decreasing noise amplitudes. This is to be expected since smaller clicks should be harder to detect than large clicks. Error probabilities may seem quite high but it should be noted that we are using Gaussian noise bursts. A significant proportion of noise samples will thus have near-zero amplitude making detection by any means a near impossibility. The dashed line gives some idea of the algorithm's sensitivity to the parameter  $\mu$ . In this case we have fixed the detector at  $\hat{\mu} = 30$  while varying the true value of  $\mu$  for the artificially generated noise bursts. Some degradation in performance can be seen which increases at low noise amplitudes (since the detector's estimate for  $\mu$  is most inaccurate for low noise burst amplitudes). Nevertheless performance is broadly similar to that for the ideal case. This result may be important in practice since  $\mu$  is

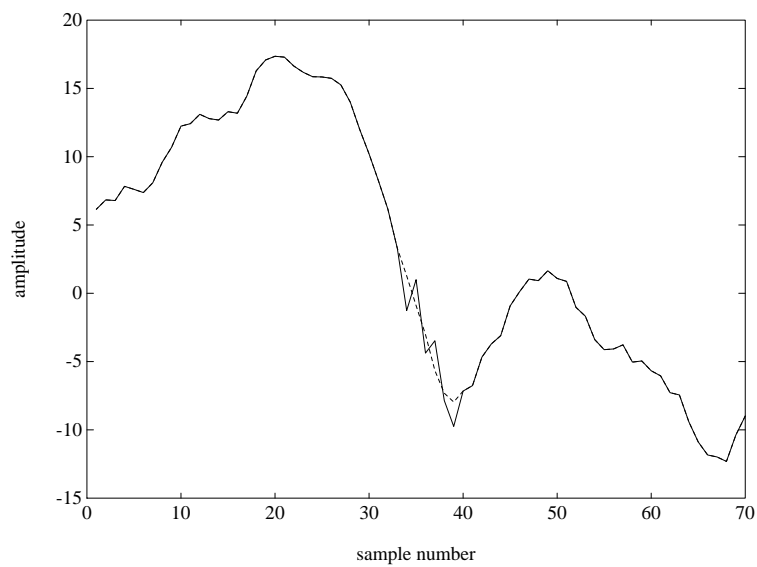


FIGURE 11.1. True audio signal (dashed line) and signal corrupted with additive Gaussian noise burst (solid line)

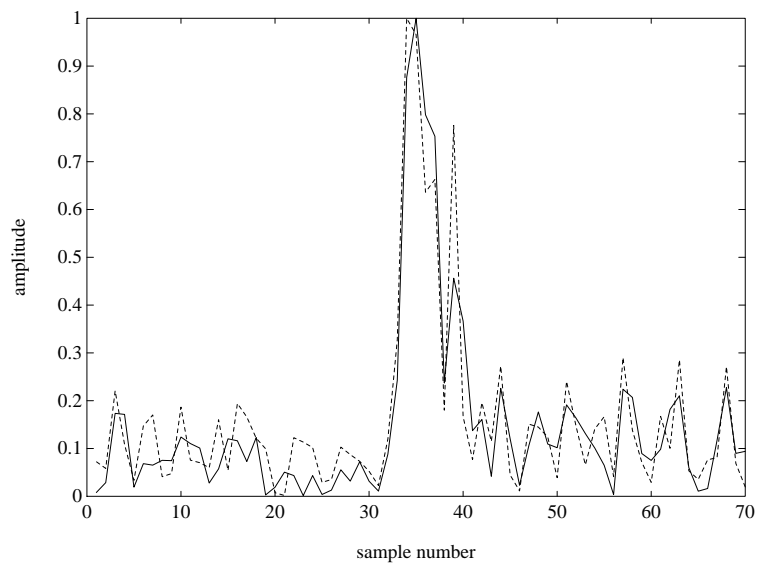


FIGURE 11.2. Detection sequences using matched (solid line) and inverse filtering (dashed line) detectors- AR(20)

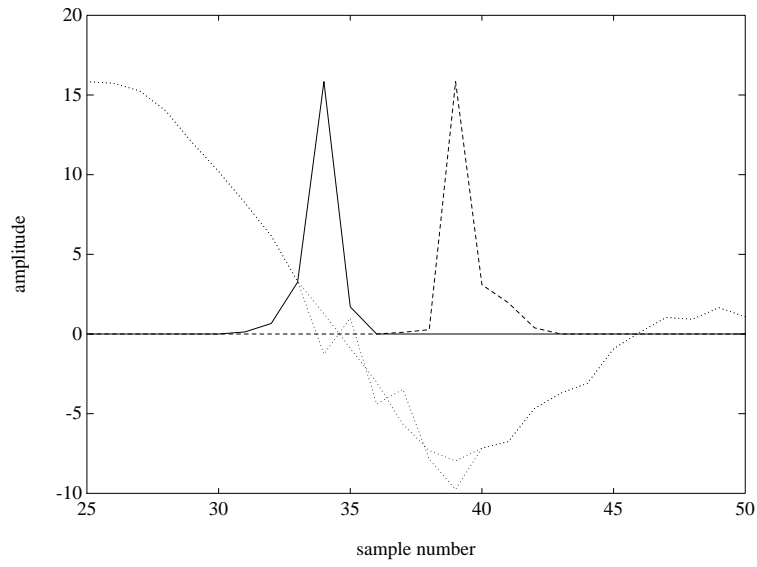


FIGURE 11.3. Posterior probabilities for  $m$  and  $n$  with AR(20) modelling: dotted line - signal waveforms; solid line - posterior probability for  $m$  with  $n$  fixed at 39; dashed line - posterior probability for  $n$  with  $m$  fixed at 31.

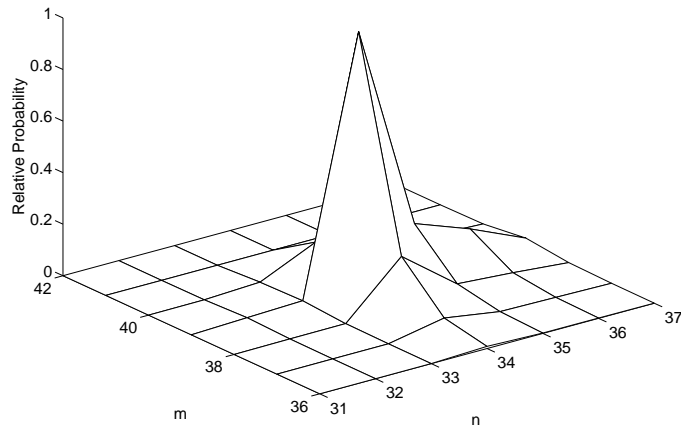


FIGURE 11.4. Mesh plot of posterior probability over a grid of  $\{m,n\}$  combinations (range is  $m:31-37$ ,  $n:36-42$ ,  $\{m,n\}$  increase vertically)

likely to be difficult to estimate. The dotted line gives the error probabilities obtained using the matched filtering detector over the same set of data samples. The threshold for this detector was hand-tuned to give minimum error probability but the result is seen to be consistently far worse than that for the Bayes detector. This result is perhaps not as poor as it appears since, while the matched filter hardly ever gives precisely the correct detection estimate owing to filter smearing effects, it is quite robust and usually gives a result close to the correct estimate. A more realistic comparison of the different methods is given later in the work on sequential detection where the *per sample* error rate is measured.

In the second graph of figure 11.6 we test the sensitivity of the detector to the Gaussian assumption for noise bursts. The noise burst amplitudes are now generated from a rectangular distribution and the detector used is based on the Gaussian assumption with  $\mu$  set according to the variance of the rectangular distribution noise samples. Performance is broadly similar to that for Gaussian noise samples with even some improvement for high noise amplitudes. The dashed line once again gives error probabilities with the estimate of  $\mu$  fixed at a value of 30.

The third graph of figure 11.7 shows results obtained using a real audio signal and artificial Gaussian noise bursts. AR parameters are estimated to order 10 by the covariance method from the uncorrupted data. Results are similar to those for the synthetic AR data indicating that the AR model assumption is satisfactory for detection in real audio signals.

Note that these experiments only give a first impression of the performance of the block-based approach since the degradation was limited to a single contiguous noise burst. A more useful objective evaluation is possible using the sequential algorithm of subsequent sections. Also, as discussed in section (9.2.3),  $P_E$  is not necessarily the best measure of the performance of this type of algorithm. For the sequential algorithms it is possible to obtain meaningful estimates of *per sample* error rates which give a more realistic picture of detector performance.

## 11.2 Sequential detection

In the last section algorithms have been proposed for block-based Bayesian detection. It has been seen that several constraints have to be placed on detection to give a practically realisable system. In particular we have seen that a good initial estimate for degraded sample locations is required. This may be unreliable or impossible to achieve especially for very low variance noise samples. The sequential algorithms derived in the last chapter allow a more flexible and general approach which does not rely on other methods or require the assumption of contiguous noise bursts which occur infrequently throughout the data.



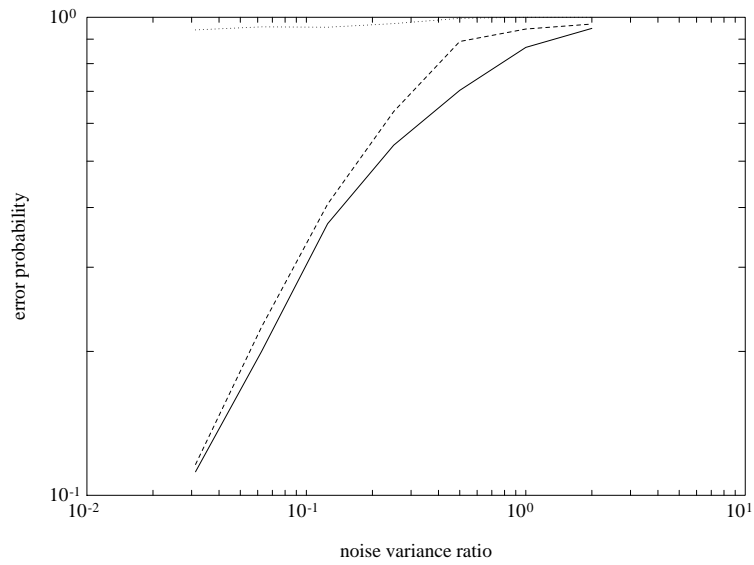


FIGURE 11.5. Error probabilities for Bayesian detector as  $\mu$  is varied, synthetic AR(10) model: solid line - Bayes with true  $\mu$  value; dashed line - Bayes with  $\hat{\mu} = 30$ ; dotted line - matched filter detector

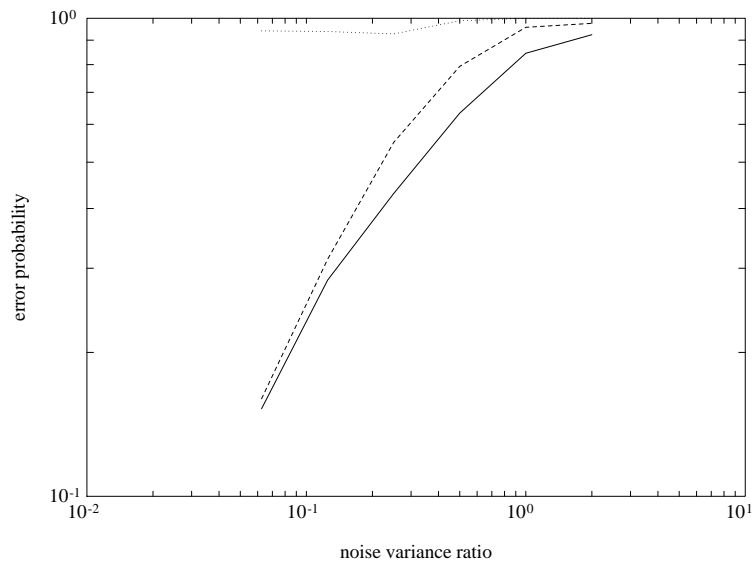


FIGURE 11.6. Error probabilities with uniform click distribution as  $\mu$  is varied, synthetic AR(10) data: solid line - Bayes with true  $\mu$  value; dashed line - Bayes with  $\hat{\mu} = 30$ ; dotted line - matched filter detector

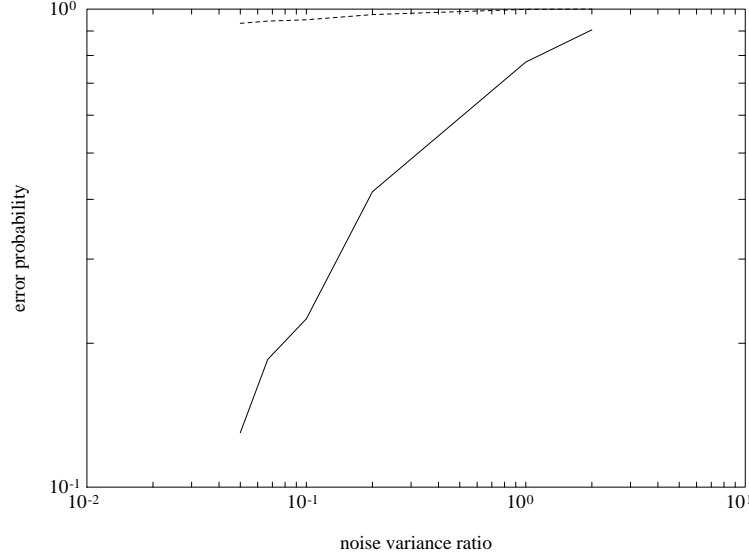


FIGURE 11.7. Error probabilities for Bayesian detector as  $\mu$  is varied, real audio data modelled as AR(10): solid line - Bayes detector; dashed line - matched filter detector

As for the block-based algorithm it is possible to make some objective evaluation of detection performance for the sequential algorithm by the processing of data corrupted with synthetically generated noise waveforms. Full knowledge of the corrupting noise process enables measurement of some critical quantities; in particular the following probabilities are defined:

$$P_f = \text{False alarm probability: Probability of detecting a corrupted sample when the sample is uncorrupted} \quad (11.2)$$

$$P_u = \text{Missed detection probability: Probability of not detecting a corrupted sample when the sample is corrupted} \quad (11.3)$$

$$P_{se} = \text{Per-sample error-rate: Probability of incorrect detection, whether false alarm or missed detection.} \quad (11.4)$$

These quantities are easily estimated in simulations by measurements of detection performance over a sufficiently large number of samples.

The first set of simulations uses a true Gaussian AR process with known parameters for the uncorrupted data signal in order to establish the algo-

rithm's performance under correct modelling assumptions. Further trials process genuine audio data for which system parameters are unknown and modelling assumptions are not perfect.

### 11.2.1 Synthetic AR data

Initially some measurements are made to determine values of the error probabilities defined in equations (11.2)-(11.4) under ideal conditions. An AR model order of 20 is used throughout. The state vector selection, or 'culling', algorithm (see section 10.3) deletes detection states which have posterior probabilities more than 50dB's below the peak value of all the candidates, and limits the maximum number of state vectors to 10 (this is the maximum value *after* the selection procedure). The hard-decision delay length is fixed at  $d_H = P$ , the AR model order. The AR model parameters for simulations are obtained by the covariance method from a typical section of audio material in order to give a realistic scenario for the simulations. The synthetic AR data is generated by filtering a white Gaussian excitation signal of known variance  $\sigma_e^2$  through the AR filter. The corrupting noise process is generated from a binary Markov source with known transition probabilities ( $P_{00} = 0.97$ ,  $P_{11} = 0.873$ ) (see section 10.2.4), and the noise amplitudes within bursts are generated from a white Gaussian noise source of known variance  $\sigma_n^2$ . The sequential detection algorithm is then applied to the synthetically corrupted data over  $N$  data samples.  $N$  is typically very large (e.g. 100000) in order to give good estimates of error probabilities. A tally is kept of the quantities  $N_f$  (the number of samples incorrectly identified as corrupted),  $N_u$  (the number of corrupted samples not detected as such) and  $N_c$  (the number of samples known to be corrupted). For sufficiently large  $N$  the error probabilities defined in (11.2)-(11.4) may be reliably estimated as:

$$\hat{P}_f = \frac{N_f}{N - N_c} \quad (11.5)$$

$$\hat{P}_u = \frac{N_u}{N_c} \quad (11.6)$$

$$\hat{P}_{se} = \frac{N_u + N_f}{N} \quad (11.7)$$

Figures (11.8) and (11.9) illustrate an example of detection and restoration within this simulation environment for  $\mu = 3$ . In the first (unrestored) waveform noise-bursts can be seen as the most 'jagged' sections in the waveform; these have been completely removed in the lower (restored) waveform.

The graphical results of figure (11.10) show the variation of detection probabilities as the ratio  $\mu$  is increase. As should be expected the probability of undetected corruption  $P_u$  increases as  $\mu$  increases and noise bursts

are of smaller amplitude. For the range of  $\mu$  examined  $P_u$  is mostly below  $10^{-1}$ , rising to higher values for very small noise amplitudes. This may seem a high proportion of undetected samples. However as noted earlier for the block-based detector noise amplitudes are generated from a Gaussian PDF, whose most probable amplitude is zero. Hence a significant proportion of noise samples will be of very low amplitude, and these are unlikely to be detected by the detector. This should not worry us unduly since such samples are least likely to leave any audible signal degradation if they remain unrestored. The rising trend in  $P_u$  is the most significant component of the overall error rate  $P_{se}$ , while  $P_f$  is consistently lower.

For comparison purposes corresponding results are plotted in the same figure for the inverse filtering detector. The error rates plotted correspond to the threshold which gives (experimentally) the minimum *per sample* error rate  $P_{se}$ . The inverse filtering detector is seen to give significantly higher error rates at all but extremely low  $\sigma_n$  values, where the two detectors converge in performance (at these extremely low noise amplitudes both detectors are unable to perform a good detection and are essentially achieving the ‘random guess’ error rate).

As for the block-based detector an important issue to investigate is the sensitivity of detector performance to incorrect parameter estimates, such as the  $\mu$  ratio and the Markov source transition probabilities  $P_{00}$  and  $P_{11}$ . Figure (11.12) shows an example of the significant variation of error probabilities as the estimate of  $\sigma_n$ ,  $\hat{\sigma}_n$ , is varied about its optimum. This variation amounts essentially to a trade-off between  $P_u$  and  $P_f$  and could be used to tailor the system performance to the needs of a particular problem. Figure (11.13) shows this trade-off in the form of a ‘Detector Operating Characteristic’ (DOC) (see [186]) with probability of correct detection ( $1 - P_u$ ) plotted against false alarm rate ( $P_f$ ). For comparison purposes an equivalent curve is plotted for the inverse filtering detector, in this case varying the detection threshold over a wide range of values. It can be seen that the Bayesian detector performs significantly better than the inverse filtering method. The shape of curve for the Bayesian method follows essentially the same shape as the standard DOC, but turns back on itself for very high  $\hat{\sigma}_n$ . This may be explained by the nature of the Bayes detector, which is now able to model high amplitude signal components as noise bursts of large amplitude, causing  $P_f$  to increase.

Considerably less performance variation has been observed when the Markov transition probabilities are varied, although once again some error-rate trade-offs can be achieved by careful choice of these parameters.

### 11.2.2 Real data

For real data it is possible to estimate the same error probabilities as for the synthetic case when noise-bursts are generated synthetically. The further issues of parameter estimation for  $\mathbf{a}$ ,  $\sigma_n$  and  $\sigma_e$  now require consideration,

since these are all assumed known by the Bayesian detector. The same principles essentially apply as in the block-based case although it is now perhaps desirable to update system parameters adaptively on a sample by sample basis. The autoregressive parameters are possibly best estimated adaptively [93] and one suitable approach is to use the *restored* waveform for the estimation. In this way the bias introduced by parameter estimation in an impulsive noise environment should be largely avoided.  $\sigma_e$  can also be estimated adaptively from the excitation sequence generated from inverse-filtering the restored data with the AR filter parameters.  $\sigma_n$  may be estimated adaptively as the root mean square value of noise samples removed during restoration.

Figure (11.11) shows the same quantities as were measured for the synthetic data case, obtained by processing a short section of wide-band music sampled from a Compact Disc recording (sample-rate 44.1kHz). Markov source probabilities and the AR model order were the same as used for the purely synthetic trials, so conditions for this real data simulation should be reasonably comparable with those results. Performance is seen to be similar to that for the synthetic AR data except for slightly higher false alarm rate  $P_f$  at low  $\mu$  values. This higher false alarm rate may reflect the fact that real data is not ideally modelled as a quasi-stationary AR process, and some transient behaviour in the data is falsely detected as degradation. The corresponding error probabilities are plotted for the inverse filtering AR detector and they are seen once more to be significantly higher.

Listening tests performed for several different sections of restored data show that an improvement in noise level reduction can be achieved by this restoration method compared with the standard AR filter-based method. In addition, the method is qualitatively judged to remove many more clicks with lower resultant signal degradation than when the inverse filtering or matched filtering detection method is used. Where restoration was performed on artificially corrupted data the restored version was compared with the true undegraded original signal. Restored sound quality was judged almost identical to that of the original material.

## 11.3 Conclusion

Implementation issues have been discussed for the Bayesian detection methods of the previous two chapters. The block-based methods have been used to evaluate performance of the Bayesian techniques and sensitivity to certain system parameters. Practical schemes for implementing block-based detection have been proposed but it has been seen that several limiting constraints must be applied in order to make the scheme usable. Sequential methods are seen to have more general application and flexibility and some useful performance measurements have been made. Substantial im-

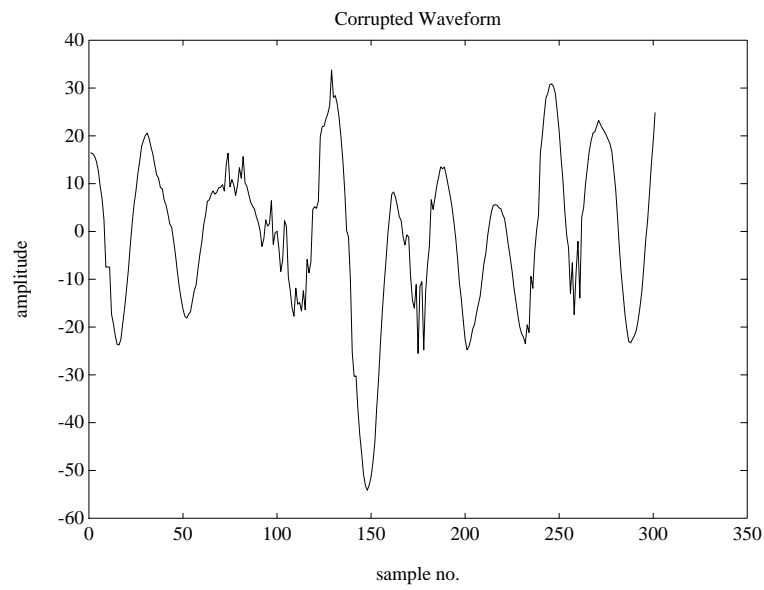


FIGURE 11.8. Corrupted input waveform for synthetic AR(20) data.

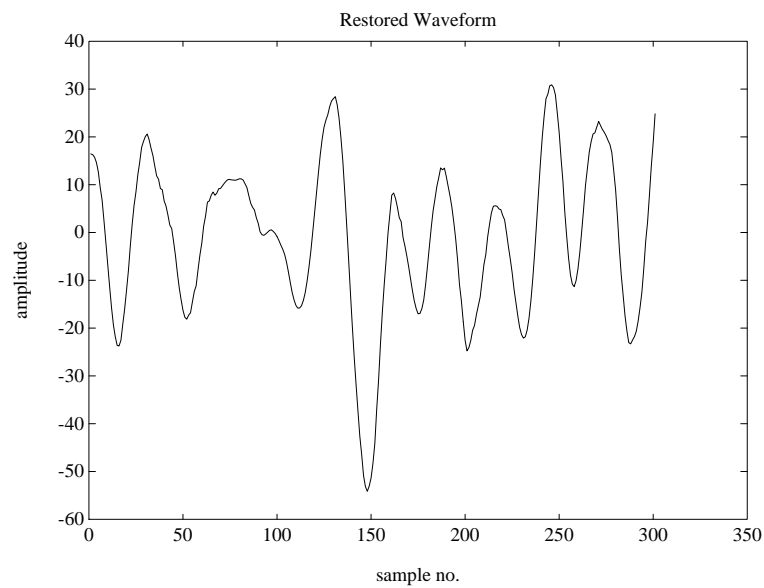


FIGURE 11.9. Restored output waveform for synthetic AR(20) data.

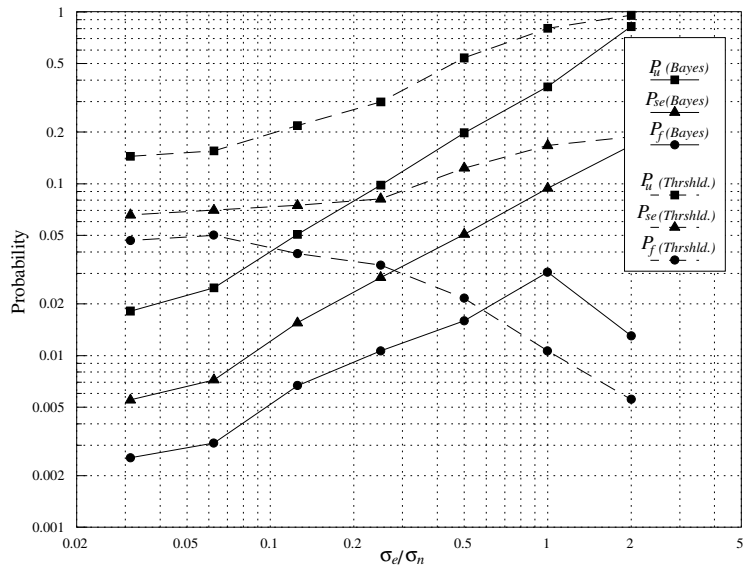


FIGURE 11.10. Error probabilities for synthetic AR(20) data as  $\frac{\sigma_e}{\sigma_n}$  is varied.

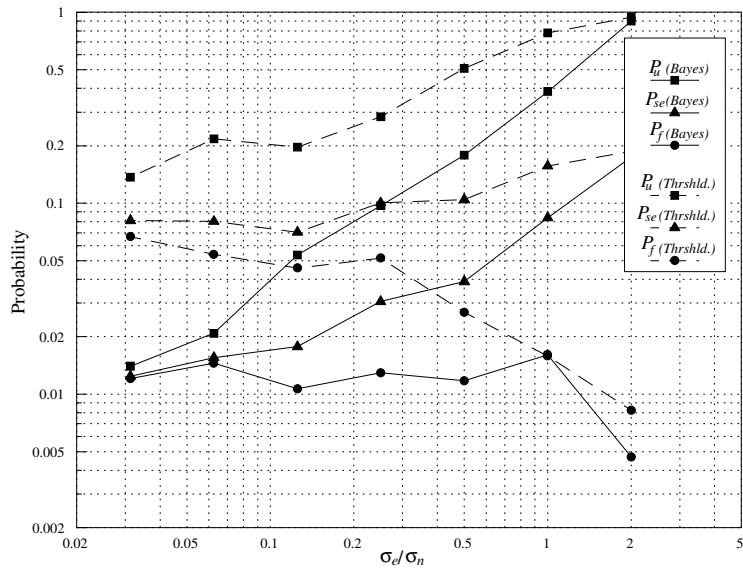


FIGURE 11.11. Error probabilities for real data (modelled as AR(20)) as  $\frac{\sigma_e}{\sigma_n}$  is varied.

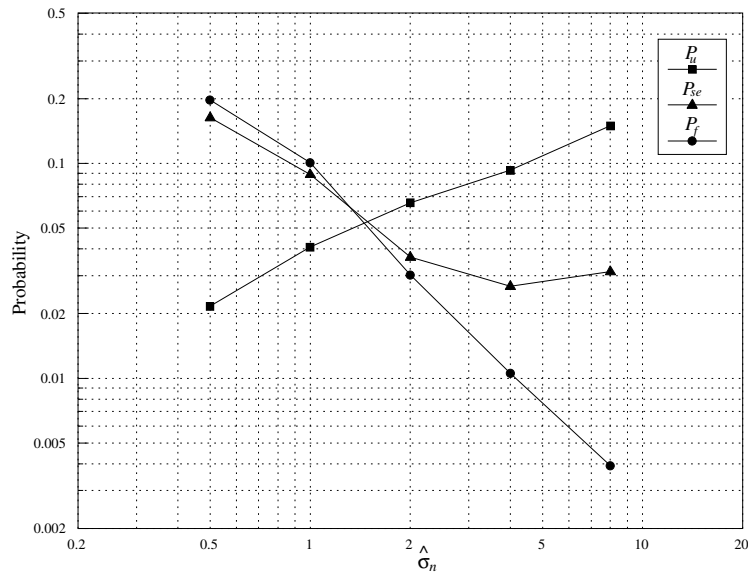


FIGURE 11.12. Error probabilities for synthetic AR(20) data as  $\hat{\sigma}_n$  is varied around the true value ( $\sigma_n = 4$ ,  $\sigma_e = 1$ ) (Bayesian method).

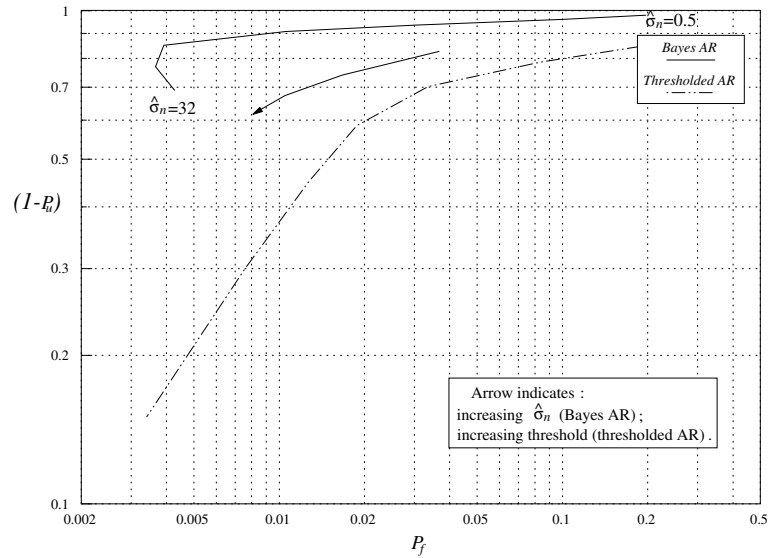


FIGURE 11.13. DOC for AR(20) data as  $\hat{\sigma}_n$  is varied around the true value ( $\sigma_n = 4$ ,  $\sigma_e = 1$ ) (Bayesian method c.f. thresholded AR.)



provements are demonstrated over the inverse filtering and matched filtering approaches to detection which indicate that the extra computational load may be worthwhile in critical applications. The Bayesian detectors tested here are, however, limited in that they assume fixed and known AR parameters, which is never the case in reality. In the final chapter we show how to lift this restriction and use numerical Bayesian methods to solve the joint problem of detection, interpolation and parameter estimation.



# 12

## Fully Bayesian Restoration using EM and MCMC

In this chapter we consider the restoration of click-degraded audio using advanced statistical estimation techniques. In the methods of earlier chapters it was generally not possible to eliminate the system hyperparameters such as the AR coefficients and noise variances without losing the analytical tractability of the interpolation and detection schemes; thus heuristic or iterative methods had to be adopted for estimation of these ‘nuisance’ parameters. The methods used here are formalised iterative procedures which allow all of the remaining hyperparameters to be numerically integrated out of the estimation scheme, thus forming estimators for just the quantities of interest, in this case the restored data. These methods, especially the Monte Carlo schemes, are highly computationally intensive and cannot currently be considered for on-line or real time implementation. However, they illustrate that fully Bayesian inference with its inherent advantages can be performed on realistic and sophisticated models given sufficient computational power. In future years, with the advancements in available computational speed which are continually being developed, methods such as these seem likely to dominate statistical signal processing when complex models are required.

The use of advanced statistical methodology allows more realistic modelling to be applied to problems, and we consider here in particular the case of non-Gaussian impulses (recall that earlier work was restricted to Gaussian impulse distributions, which we will see can lead to inadequate performance). Modelling of non-Gaussianity is achieved here by use of *scale mixtures of Gaussians*, a technique which allows quite a wide range of heavy-tailed noise distributions to be modelled, and some detailed dis-

cussion of this important point is given. Having described the modelling framework for clicks and signal, an expectation-maximisation (EM) algorithm is presented for the interpolation of corrupted audio in the presence of non-Gaussian impulses. This method provides effective results for the pure interpolation problem (i.e. when the detection vector  $\mathbf{i}$  is known) but cannot easily be extended to detection as well as interpolation. The Gibbs sampler provides a method for achieving this, and the majority of the chapter is devoted to a detailed description of a suitable implementation of this scheme. Finally results are presented for both methods. The work included here has appeared previously as [80, 82, 83].

## 12.1 A review of some relevant work from other fields in Monte Carlo methods

Applications of MCMC methods which are of particular relevance to this current work include Carlin *et al.* [33] who perform MCMC calculations in a general non-linear and non-Gaussian state-space model, McCulloch and Tsay [128, 129] who use the Gibbs Sampler to detect change-points and outliers in autoregressive data and, in the audio processing field, Ó Ruanaidh and Fitzgerald [146] who develop interpolators for missing data in autoregressive sequences.

Take first Carlin *et al.* [33]. This paper takes a non-linear, non-Gaussian state-space model as its starting point and uses a rejection-based MCMC method to sample the conditional for the state variables at each sampling time in turn. As such this is a very general formulation. However, it does not take into account any specific state-space model structure and hence could converge poorly in complex examples. Our method by contrast expresses the model for noise sources using extra latent variables (in particular the indicator variables  $i_t$  and the noise variances  $\sigma_{v_t}^2$ , see later) which leads to a *conditionally Gaussian* structure [36, 170] for the reconstructed data  $\mathbf{x}$  and certain model parameters. This conditionally Gaussian structure is exploited to create efficient blocking schemes in the Gibbs Sampler [113] which are found to improve convergence properties.

The signal and noise models we use are related to those used by us in chapter 9 and [76, 83] and also McCulloch and Tsay [129] for analysis of autoregressive (AR) time series. Important differences between our MCMC schemes and theirs include the efficient blocking schemes we use to improve convergence (McCulloch and Tsay employ only univariate conditional sampling), our model for the continuous additive noise source  $v_t$ , which is based on a continuous mixture of Gaussians in order to give robustness to non-Gaussian heavy-tailed noise sources, and in the discrete Markov chain prior which models the ‘burst’-like nature of typical impulsive processes. A further development is treatment of the noise distribution ‘hyperparameters’

as unknowns which can be sampled alongside the other parameters. This allows both the scale parameter and the degrees of freedom parameter for the impulsive noise to be estimated directly from the data.

Finally, the Gibbs Sampling and EM schemes of Ó Ruanaidh and Fitzgerald [146, 168] consider the interpolation of missing data in autoregressive sequences. As such they do not attempt to model or detect impulsive noise sources, assuming rather that this has been achieved beforehand. Their work can be obtained as a special case of our scheme in which the noise model and detection steps (both crucial parts of our formulation) are omitted.

## 12.2 Model specification

### 12.2.1 Noise specification

The types of degradation we are concerned with here can be regarded as additive and localised in time, which may be represented quite generally using a ‘switched’ noise model as seen previously in chapters 5 and 9:

$$y_t = x_t + i_t v_t \quad (12.1)$$

where  $y_t$  is the observed (corrupted) waveform,  $x_t$  is the underlying audio signal and  $i_t v_t$  is an impulsive noise process.  $i_t$  is a binary (0/1) ‘indicator’ variable which indicates outlier positions and  $v_t$  is a continuous noise process.

#### 12.2.1.1 Continuous noise source

An important consideration within a high fidelity reconstruction environment will be the statistics of the continuous noise source,  $v_t$ . Close examination of the typical corrupted audio waveform in figure 12.15 indicates that the noise process operates over a very wide range of amplitudes within a short time interval. In the case of gramophone recordings this is a result of physical defects on the disk surface which vary dramatically in size, from microscopic surface cracks up to relatively large dust particles adhering to the groove walls and scratches in the medium, while in a communications environment variations in the size of impulses can be attributed to the power of individual noise sources and to their distance from the measuring point. It will clearly be important to model the noise sources in a fashion which is robust to the full range of defect sizes which is likely to be encountered. In particular, for very small-scale defects it should be possible to extract useful information about the underlying signal from corrupted sample values, while very large impulses should effectively be ignored and treated as if the data were ‘missing’.

A natural first assumption for the underlying noise process  $v_t$  might be the Gaussian distribution  $p(v_t|\sigma_v^2) = \mathcal{N}(0, \sigma_v^2)$ . Such an assumption has been investigated for degraded audio signals within a Bayesian framework in [76, 79] and chapters 9-11. However, the standard Gaussian assumption is well known to be non-robust to noise distributions which are more ‘heavy-tailed’ than the Gaussian, even if the variance component  $\sigma_v^2$  is made an unknown parameter *a priori*.

A convenient way to make the noise model robust to heavy-tailed noise sources is to retain the overall Gaussian framework but to allow the variance components of individual noise sources to vary with time index. In this way a reconstruction algorithm can adjust *locally* to the *scale* of individual defects, while the computational advantages of working within a linear Gaussian framework are retained. The noise source  $v_t$  is thus modelled as Gaussian with time-varying variance parameter, i.e.  $p(v_t|\sigma_{v_t}^2) = \mathcal{N}(0, \sigma_{v_t}^2)$  where  $\sigma_{v_t}^2$  is dependent upon the time index  $t$ .

Under these assumptions the likelihood  $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$  is then obtained from the independent, switched, additive noise modelling considerations (see equation (12.1)). Note that  $\boldsymbol{\theta}$  contains all the system unknowns. However, since the noise source is assumed to be independent of the signal, the likelihood is conditionally independent of the signal model parameters. Define as before  $\mathbf{i}$  to be the vector of  $i_t$  values corresponding to  $\mathbf{x}$ . Then,

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{\{t:i_t=0\}} \delta(y_t - x_t) \times \prod_{\{t:i_t=1\}} \mathcal{N}(y_t|x_t, \sigma_{v_t}^2) \quad (12.2)$$

where the  $\delta$ -functions account for the fact that the additive noise source is precisely zero whenever the indicator switch  $i_t$  is zero (see (12.1)). Note that the likelihood depends only upon the noise parameters (including  $\mathbf{i}$ ) and is independent of any other elements in  $\boldsymbol{\theta}$ .

### 12.2.2 Signal specification

Many real-life signals, including speech, music and other acoustical signals, have strong local autocorrelation structure in the time domain. This structure can be used for distinguishing between an uncorrupted audio waveform and unwanted noise artefacts. We model the autocorrelation structure in the uncorrupted data sequence  $\{x_t\}$  as an autoregressive (AR) process (as in earlier chapters) whose coefficients  $\{a_i\}$  are constant in the short term:

$$x_t = \sum_{i=1}^P x_{t-i} a_i + e_t \quad (12.3)$$

where  $e_t \sim N(0, \sigma_e^2)$  is an i.i.d. excitation sequence. In matrix-vector form we have, for  $N$  data samples:

$$\mathbf{e} = \mathbf{A}\mathbf{x} = \mathbf{x}_1 - \mathbf{X}\mathbf{a} \quad (12.4)$$

where the rows of  $\mathbf{A}$  and  $\mathbf{X}$  are constructed in such a way as to form  $e_t$  in (12.3) for successive values of  $t$  and the notation  $\mathbf{x}_1$  denotes vector  $\mathbf{x}$  with its first  $P$  elements removed. The parametric model for signal values conditional upon parameter values can now be expressed using the standard approximate likelihood for AR data (section 4.3.1 and [21, 155]):

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{a}, \sigma_e^2) = N_N(\mathbf{0}, \sigma_e^2(\mathbf{A}^T \mathbf{A})^{-1}) \quad (12.5)$$

where  $N_q(\boldsymbol{\mu}, \mathbf{C})$  denotes as before the multivariate Gaussian with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\mathbf{C}$  (see appendix A).

It is of course an approximation to assume that the AR parameters and excitation variance remain fixed in any given block of data. A time-varying system might be a more realistic representation of the signal and should certainly be considered in future adaptations to this work (see, e.g. [160] for a possible framework). However, we have found this assumption to be much less critical in practice than the choice of an appropriate impulsive noise model (see section 12.8.1 for comparison with a constant variance Gaussian impulse model), since the signal parameters for typical speech and audio signals typically vary slowly and smoothly with time. A further approximation is the assumption of Gaussian excitation process  $\{e_t\}$ , which may not hold well for many speech or music examples. It is possible to relax this assumption and allow impulsive elements in the excitation sequence. This issue is not investigated here, since we have found that results are generally robust to the Gaussian assumption. However, some initial work in this area can be found in [72, 81].

## 12.3 Priors

### 12.3.1 Prior distribution for noise variances

Within a Bayesian framework prior distributions can now be assigned to the unknown variance components  $\sigma_{v_t}^2$ . In principle any distribution which is defined over the positive axis can be chosen, and the precise choice will depend on the type of prior knowledge available (if any) and will lead to different robustness properties of the reconstruction procedure. A few possible distributions which express different forms of prior knowledge are

listed below:

$$\begin{aligned}
p(\sigma_{v_t}^2) &\propto c, && \text{Uniform} \\
p(\sigma_{v_t}^2) &\propto 1/\sigma_{v_t}^2, && \text{Jeffreys} \\
p(\log(\sigma_{v_t}^2)) &= \mathcal{N}(\mu_v, s_v), && \text{Log-normal} \\
p(\sigma_{v_t}^2) &= \text{IG}(\alpha_v, \beta_v) \propto (\sigma_{v_t}^2)^{-(\alpha_v+1)} \exp(-\beta_v/\sigma_{v_t}^2), && \text{Inverted-gamma}
\end{aligned}$$

Uniform or Jeffreys [97] priors might be used when no prior information is available about noise variances. The log-normal prior is a suitable choice when expected noise levels are known or estimable *a priori* on a decibel scale. The inverted-gamma (IG) prior [98] (see appendix A) is a convenient choice in that vague or specific prior information can be incorporated through suitable choice of  $\alpha_v$  and  $\beta_v$  while the uniform and Jeffreys priors can be obtained as improper limiting cases. Owing to its flexibility and computational simplicity (see later) the IG prior will be adopted for the remainder of this work. Note that upper and lower limits on the noise variance may well be available from physical considerations of the system and could easily be incorporated by rejection sampling [46, 153] into the Gibbs sampling scheme proposed below.

We have discussed intuitively why combinations of noise models and priors such as these can be expected to lead to robustness. Perhaps a more concrete interpretation is given by studying the marginal distribution  $p(v_t)$  which is obtained when the variance term is integrated out, or marginalised, from the joint distribution  $p(v_t|\sigma_{v_t}^2)p(\sigma_{v_t}^2)$ . The resulting distributions are *scale mixtures of normals*, which for example West [196] has proposed for modelling of heavy-tailed noise sources. The family of possible distributions which can result from this scale mixture process is large, including such well-known robust distributions as  $\alpha$ -stable, generalised Gaussian and Student-t families [7, 197]. In fact, the Student-t distribution is known to arise as a result of assuming the IG prior (see appendix H.1 for a derivation), and we adopt this as the prior density in the remainder of the chapter, owing both to its convenient form computationally and its intuitive appeal. The Student-t distribution is a well known robust distribution which has frequently been proposed for modelling of impulsive sources (see e.g. [89]). However, it should be noted that the sampling methods described here can be extended to other classes of prior distribution on the noise variances in cases where Student-t noise is not thought to be appropriate. This will lead to other robust noise distributions such as the  $\alpha$ -stable class [144] which has received considerable attention recently as the natural extension to a Gaussian framework.  $\alpha$ -stable noise distributions, being expressible as a scale mixture of normals [7, 197], can in principle fit into the same framework as we propose here, although the lack of an analytic form for the distribution complicates matters somewhat.



### 12.3.2 Prior for detection indicator variables

The statistics of the continuous noise source  $v_t$  have been fully specified. It now remains to assign a prior distribution to the vector  $\mathbf{i}$  of noise switching values  $i_t$ . The choice of this prior is not seriously limited by computational considerations since any prior which can easily be evaluated for all  $\mathbf{i}$  will fit readily into the sampling framework of the next section. Since it is known that impulses in many sources occur in ‘bursts’ rather than singly (see [70, 79] for discussion of the audio restoration case), we adopt a two-state Markov chain prior to model transitions between adjacent elements of  $\mathbf{i}$ . A model of this form has previously been proposed in [67] for burst-noise in communications channels. The prior may be expressed in the form  $p(\mathbf{i}) = p(i_0) \prod_t P_{i_{t-1}i_t}$ , in which the clustering of outliers in time is modelled by the transition probabilities of the Markov chain,  $P_{ij}$  and  $p(i_0)$  is the prior on the initial state. These transition probabilities can be assigned *a priori* based on previous experience with similar data sets, although they could in principle be ‘learnt’ by the Gibbs Sampler along with the other unknowns (see later).

### 12.3.3 Prior for signal model parameters

Assuming that the AR parameter vector  $\mathbf{a}$  is *a priori* independent of  $\sigma_e^2$ , we assign the simple improper prior  $p(\mathbf{a}, \sigma_e^2) = p(\mathbf{a})p(\sigma_e^2) \propto \text{IG}(\sigma_e^2 | \alpha_e, \beta_e)$ , with the parameters  $\alpha_e$  and  $\beta_e$  chosen so that the IG distribution approaches the Jeffreys limiting case. Note that a very vague prior specification such as this should be adequate in most cases since  $\mathbf{a}$  and  $\sigma_e^2$  will be well-determined from the data, which will contain many hundreds of elements. A slightly more general framework would allow for a proper Gaussian prior on  $\mathbf{a}$ . We do not include this generalisation here for the sake of notational simplicity, but it should be noted that the general mathematical form of the Gibbs sampler derived shortly remains unchanged under this more general framework.

## 12.4 EM algorithm

We firstly consider the application of the EM algorithm (see section 4.5) to the pure interpolation problem, i.e. estimation of  $\mathbf{x}_{(i)}$  conditional upon the detection indicator vector  $\mathbf{i}$  and the observed data  $\mathbf{y}$ . In this section we work throughout in terms of *precision* parameters rather than variance parameters, i.e.  $\lambda = 1/\sigma^2$ , assigning gamma priors rather than inverted gamma. This is purely for reasons of notational simplicity in the derivation, and a simple change of variables  $\lambda = 1/\sigma^2$  gives exactly equivalent results based on the corresponding variance parameters. We will revert to the variance form in the subsequent Gibbs sampler interpolation algorithm.

For purposes of comparison, we present the EM algorithm for both the assumed noise model above, with independent variance components for each noise sample, and a *common variance* model as used in chapter 9 where the noise samples are Gaussian with a single common variance term  $\sigma_v^2$ .

Substituting  $\lambda_e = 1/\sigma_e^2$ , the conditional AR likelihood (12.5) is rewritten as

$$p(\mathbf{x}|\mathbf{a}, \lambda_e) = (\lambda_e/2\pi)^{\frac{N-P}{2}} \exp(-\lambda_e E(\mathbf{x}, \mathbf{a})/2) \stackrel{def}{=} \text{AR}_N(\mathbf{x}|\mathbf{a}, \lambda_e) \quad (12.6)$$

where  $\lambda_e = 1/\sigma_e^2$  and  $E(\mathbf{x}, \mathbf{a}) = \mathbf{e}^T \mathbf{e}$  is the squared norm of the vector of excitation values.

The interpolation problem is posed in the usual way as follows. A signal  $\mathbf{x}$  is corrupted by additive noise affecting  $l$  specified elements of  $\mathbf{x}$ , indexed by a switching vector  $\mathbf{i}$ . The data  $\mathbf{x}$  is partitioned according to ‘unknown’ (noise-corrupted) samples  $\mathbf{x}_{(i)}$  whose time indices form a set  $\mathcal{I}$ , and the remaining ‘known’ (uncorrupted) samples, denoted by  $\mathbf{x}_{-(i)}$ . The observed data is  $\mathbf{y}$ , and by use of a similar partitioning we may write  $\mathbf{y}_{-(i)} = \mathbf{x}_{-(i)}$  and  $\mathbf{y}_{(i)} = \mathbf{x}_{(i)} + \mathbf{v}_{(i)}$ , where  $\mathbf{v} = [v_1, \dots, v_N]^T$  is the corrupting noise signal. It is then required to reconstruct the unknown data  $\mathbf{x}_{(i)}$ . Note that the noise variance, the AR coefficients and the AR excitation variance will all be unknown in general, so these will be treated as model parameters, denoted by  $\boldsymbol{\theta}$ . We use an unconventional formulation of EM in which the parameters  $\boldsymbol{\theta}$  are treated as the auxiliary data and the missing data  $\mathbf{x}_{(i)}$  is treated as the quantity of interest, i.e. we are performing the following MAP estimation:

$$\mathbf{x}_{(i)}^{\text{MAP}} = \underset{\mathbf{x}_{(i)}}{\text{argmax}} \{p(\mathbf{x}_{(i)}|\mathbf{x}_{-(i)})\} = \underset{\mathbf{x}_{(i)}}{\text{argmax}} \left\{ \int_{\boldsymbol{\theta}} p(\mathbf{x}_{(i)}, \boldsymbol{\theta}|\mathbf{x}_{-(i)}) d\boldsymbol{\theta} \right\}$$

This is the formulation used by ÓRuanaidh and Fitzgerald [168] for the purely missing data problem (i.e. with no explicit noise model) but is distinct from say [192, 136], who use EM for estimation of parameters rather than missing data. Here the operation of EM is very powerful, enabling us to marginalise all of the nuisance parameters, including the AR model and the noise statistics.

The derivation of the EM interpolator is given in appendix G. It requires calculation of expected values for the signal and noise parameters, from which the updated missing data estimate  $\mathbf{x}_{(i)}^{i+1}$  (iteration  $i+1$ ) can be obtained from the current estimate  $\mathbf{x}_{(i)}^i$  (iteration  $i$ ), for the independent

noise variance model, as follows:

$$\mathbf{a}^i = \mathbf{a}^{\text{MAP}}(\mathbf{x}^i) \quad (12.7)$$

$$\lambda_e^i = \frac{(\alpha_e + (N - P)/2)}{(\beta_e + E(\mathbf{x}^i, \mathbf{a}^i)/2)}$$

$$\lambda_{v_t}^i = (\alpha_v + 1/2)/(\beta_v + (v_t^i)^2/2), \quad (t \in \mathcal{I}) \quad (12.8)$$

$$\mathbf{x}_{(i)}^{i+1} = -\Psi^{-1} \left( \left( \lambda_e^i \mathbf{A}^{iT} \mathbf{A}^i + \mathbf{T}(\mathbf{x}^i) \right)_{(i)-(i)} \mathbf{y}_{-(i)} - \mathbf{M}^i \mathbf{y}_{(i)} \right) \quad (12.9)$$

where

$$\mathbf{a}^{\text{MAP}}(\mathbf{x}^i) = (\mathbf{X}^{iT} \mathbf{X}^i)^{-1} \mathbf{X}^{iT} \mathbf{x}_1^i \quad (12.10)$$

$$\mathbf{M}^i = \text{diag}(\{\lambda_{v_t}; t \in \mathcal{I}\})$$

$$\Psi = \left( \lambda_e^i \mathbf{A}^{iT} \mathbf{A}^i + \mathbf{T}(\mathbf{x}^i) \right)_{(i)(i)} + \mathbf{M}^i$$

Matrix  $\mathbf{T}(\mathbf{x}^i)$  is defined in appendix G and ‘ $_{(i)(i)}$ ’ denotes the sub-matrix containing all elements whose row and column numbers are members of  $\mathcal{I}$ ; similarly, ‘ $_{(i)-(i)}$ ’ extracts the sub-matrix whose row numbers are in  $\mathcal{I}$  and whose column numbers are not in  $\mathcal{I}$ . The common variance model is obtained simply by replacing the observation noise expectation step (12.8) with

$$\lambda_v^i = (\alpha_n + l/2)/(\beta_v + \sum_{t \in \mathcal{I}} (v_t^i)^2/2)$$

and setting  $\mathbf{M}^i = \lambda_v^i \mathbf{I}$ .

The first three lines of the iteration are simply calculating expected values of the unknown system parameters (AR coefficients and noise precision values) conditional upon the current missing data estimate  $\mathbf{x}_{(i)}^i$ . These are then used in the calculation of  $\mathbf{x}_{(i)}^{i+1}$ . Each iteration of the method thus involves a sequence of simple linear estimation tasks. Each iteration is guaranteed to increase the posterior probability of the interpolated data. Initialising with a starting guess  $\mathbf{x}_{(i)}^0$ , the algorithm is run to convergence according to some suitable stopping criterion.

In section 12.6 some results for the EM interpolator are presented and compared with the Gibbs sampling scheme. We now consider the Gibbs sampler both for the interpolation problem considered here and also for the more challenging case of joint detection and interpolation of click-degraded data.

## 12.5 Gibbs sampler

The noise and signal probability expressions given in the previous sections are sufficient to specify the joint posterior distribution to within a con-

stant scale factor (see appendix H.2). The Gibbs sampler now requires the full posterior conditional distributions for each unknown in turn. The standard approach in many applications is to calculate univariate conditionals for each parameter component in turn. This is the approach given in, for example, [33] for non-linear/non-Gaussian state-space analysis, and also in [129] for a statistical outlier model which is closely related to our switched noise model. This sampling approach offers generality in that it is usually feasible to sample from the univariate conditionals even for very sophisticated non-Gaussian models. However, as noted earlier, improved convergence can be expected through judicious choice of multivariate parameter subsets in the sampling steps. In particular, for the models we have chosen it is possible to sample *jointly* from the switching vector  $\mathbf{i}$  and the reconstructed data vector  $\mathbf{x}$ .

### 12.5.1 Interpolation

Consider firstly the conditional densities with the switching vector  $\mathbf{i}$  fixed and known. Since no detection process is required to determine which data samples are corrupted this may be considered as a pure *interpolation* procedure in which the corrupted samples are replaced by sampled realisations of their uncorrupted value. The methods outlined in this section could of course be used as a stand-alone interpolation technique in cases where impulse locations are known beforehand, although we will usually choose here to perform the interpolation and detection (see next section) jointly within the Gibbs sampling framework.

From this point on we split  $\boldsymbol{\theta}$  into the switching vector  $\mathbf{i}$  and remaining model parameters  $\boldsymbol{\omega} = \{\mathbf{a}, \sigma_e^2, \sigma_{v_t}^2 \ (t = 0 \dots N-1)\}$  for the sake of notational clarity. The posterior conditionals are then obtained by straightforward manipulations of the full joint posterior (see appendix H.2) and are summarised as:

$$p(\mathbf{a}|\mathbf{x}, \mathbf{i}, \boldsymbol{\omega}_{-(\mathbf{a})}, \mathbf{y}) = N_P(\mathbf{a}^{\text{MAP}}(\mathbf{x}), \sigma_e^2 (\mathbf{X}^T \mathbf{X})^{-1}) \quad (12.11)$$

$$p(\sigma_e^2|\mathbf{x}, \mathbf{i}, \boldsymbol{\omega}_{-(\sigma_e^2)}, \mathbf{y}) = \text{IG}(\alpha_e + (N - P)/2, \beta_e + E(\mathbf{x}, \mathbf{a})/2) \quad (12.12)$$

$$p(\sigma_{v_t}^2|\mathbf{x}, \mathbf{i}, \boldsymbol{\omega}_{-(\sigma_{v_t}^2)}, \mathbf{y}) = \begin{cases} \text{IG}(\alpha_v + 1/2, \beta_v + v_t^2/2) & (t \in \mathcal{I}) \\ \text{IG}(\alpha_v, \beta_v) & (\text{otherwise}) \end{cases} \quad (12.13)$$

$$p(\mathbf{x}_{(\mathbf{i})}|\mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) = N_l(\mathbf{x}_{(\mathbf{i})}^{\text{MAP}}, \sigma_e^2 \boldsymbol{\Phi}^{-1}) \quad (12.14)$$

$$\mathbf{x}_{-(\mathbf{i})} = \mathbf{y}_{-(\mathbf{i})} \quad (12.15)$$

where

$$\begin{aligned}\mathbf{a}^{\text{MAP}}(\mathbf{x}) &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{x}_1 \\ \mathbf{x}_{(i)}^{\text{MAP}} &= -\Phi^{-1} \left( \mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)} - \sigma_e^2 \mathbf{R}_{\mathbf{v}_{(i)}}^{-1} \mathbf{y}_{(i)} \right) \\ \Phi &= \mathbf{A}_{(i)}^T \mathbf{A}_{(i)} + \sigma_e^2 \mathbf{R}_{\mathbf{v}_{(i)}}^{-1}\end{aligned}$$

and  $\omega_{-(\mathbf{a})}$  denotes all members of  $\omega$  except  $\mathbf{a}$ , etc. A similar notation for vector/matrix partitioning to that used for the EM interpolator and earlier chapters should be clear from the context; defined such that subscript ‘ $(i)$ ’ denotes elements/columns corresponding to corrupted data (i.e.  $i_t = 1$ ), while ‘ $-(i)$ ’ denotes the remaining elements/columns. The terms  $v_t$ , required (12.13) are calculated for a particular  $\mathbf{x}$ ,  $\mathbf{y}$  and  $\mathbf{i}$  from (12.1).  $\mathbf{R}_{\mathbf{v}_{(i)}}$  is the covariance matrix for the  $l$  corrupting impulses indicated by  $\mathbf{i}$ , and is diagonal with elements  $\sigma_{v_t}^2$  in this case. Note also that when  $i_t = 0$ , (12.13) requires that we sample from the prior on  $\sigma_{v_t}^2$ , since there is no information available from the data or parameters about this hyperparameter when the impulsive noise source is switched ‘off’. It is nevertheless important to carry out this sampling operation since the detection procedure (see next section) requires sampled values of  $\sigma_{v_t}^2$  for all  $t$ .

An important point about the method is that all of the sampling steps (12.11)-(12.14) involve only simple linear operations and sampling from multivariate normal and inverted-gamma distributions, for which reliable and fast methods are readily available (see e.g. [26, 46]). The scheme involves multivariate sampling of the reconstructed data  $\mathbf{x}$  in order to achieve reliable convergence properties, in the same spirit as [161, 79, 146], and indeed the Gibbs sampling interpolator given in [146] can be derived as a special case of (12.11)-(12.14) when the corrupted samples are discarded as ‘missing’.

#### 12.5.1.0.1 Sampling the noise hyperparameters

The above working is all implicitly conditioned upon knowledge of the noise hyperparameters  $\alpha_v$  and  $\beta_v$ . Within the Gibbs sampling framework there is no reason in principle why we should not assign priors to these parameters and treat them as unknowns. The required conditional densities and sampling operations can be obtained straightforwardly and are summarised in appendix H.3, along with the form of prior distributions chosen. Whether these variables should be treated as unknowns will depend on the extent of prior information available. For example, when processing very long sections of data such as speech or music extracts it will be advisable to split the data into manageable sub-blocks in which the signal model parameters can be regarded as constant. However, in such cases it is often reasonable to assume that the highest level noise parameters  $\alpha_v$  and  $\beta_v$  remain roughly constant for all blocks within a given extract. Hence it might be best to ‘learn’ these parameters informally through the processing of earlier data

blocks rather than starting afresh in each new data block with very vague prior information. Such a procedure certainly speeds up convergence of the Gibbs Sampler, which can take some time to converge if  $\alpha_v$  and  $\beta_v$  are randomly initialised and ‘vague’ priors are used. In order to demonstrate the full capabilities of the methods, however, the results presented later treat  $\alpha_v$  and  $\beta_v$  as unknowns which are then sampled with the remaining variables.

### 12.5.2 Detection

Now consider the remaining unknown, the switch vector  $\mathbf{i}$ . Sampling for  $\mathbf{i}$  as part of the Gibbs Sampling scheme will allow joint *detection* of impulse locations as well as reconstruction of clean data values. A straightforward Gibbs Sampling approach will sample from the conditional for each univariate  $i_t$  element. This is the approach adopted in [129] and it is similar in principle to the variable selection methods for linear regression given in [65].

In [129] univariate sampling is performed from the conditionals for all the elements  $i_t$  and  $v_t$  (from which the sampled reconstruction can be obtained as  $x_t = y_t - i_t v_t$ ). This scheme has two drawbacks which will affect convergence of the sampler. Firstly, sampling is univariate even though there are likely to be strong posterior correlations between successive elements of  $i_t$  and  $v_t$ . Secondly, the scheme involves sampling from the prior for  $v_t$  when  $i_t = 0$ , since  $v_t$  is conditionally independent of the input data in this situation.  $i_t$  will then only stand a reasonable chance of detecting a true outlier when  $v_t$  happens to take a sampled value close to its ‘true’ value, with the result that the evolution of  $i_t$  with iteration number is rather slow. Note that in equation (12.13) of our scheme it is necessary to sample from the prior on  $\sigma_{v_t}^2$  when  $i_t = 0$ . However, at this level of the ‘hierarchy’ the convergence of  $i_t$  is less likely to be affected adversely. The drawbacks of univariate sampling can be overcome by sampling from the joint multivariate conditional for  $\mathbf{x}$  and  $\mathbf{i}$ . Note that this is distinct from the approach of Carter and Kohn for estimation of mixture noise [35], in which indicator variables ( $\mathbf{i}$ ) are sampled jointly *conditional* upon state variables ( $\mathbf{x}$ ), and *vice versa*.

The proposed scheme uses the joint multivariate conditional distribution for  $\mathbf{x}$  and  $\mathbf{i}$ , which can be factorised as:

$$p(\mathbf{x}, \mathbf{i} | \boldsymbol{\omega}, \mathbf{y}) = p(\mathbf{x} | \mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) p(\mathbf{i} | \boldsymbol{\omega}, \mathbf{y}) \quad (12.16)$$

The first term of this has already been given in (12.14). The second term, the so-called *reduced conditional*, is the conditional distribution for  $\mathbf{i}$  and  $\mathbf{x}$  with  $\mathbf{x}$  ‘integrated out’:

$$p(\mathbf{i} | \boldsymbol{\omega}, \mathbf{y}) = \int_{\mathbf{x}} p(\mathbf{x}, \mathbf{i} | \boldsymbol{\omega}, \mathbf{y}) d\mathbf{x} \quad (12.17)$$

This integral can be performed directly using multivariate normal probability identities as in [79, equations (10)-(15)] and section 9.2.2.3 of this book and may be expressed as

$$p(\mathbf{i}|\boldsymbol{\omega}, \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\omega}, \mathbf{i}) p(\mathbf{i})$$

where

$$p(\mathbf{y}|\boldsymbol{\omega}, \mathbf{i}) = \frac{(\sigma_e^2)^{l/2} \exp(-S^2|_{\mathbf{x}_{(i)}=\mathbf{x}_{(i)}^{\text{MAP}}})}{(2\pi\sigma_e^2)^{(N-P)/2} |\mathbf{R}_{\mathbf{v}_{(i)}}|^{1/2} |\boldsymbol{\Phi}|^{1/2}} \quad (12.18)$$

and

$$S^2 = \sum_t e_t^2 / 2\sigma_e^2 + \sum_{\{t: i_t=1\}} v_t^2 / 2\sigma_{v_t}^2 \quad (12.19)$$

Note that  $S^2$  in (12.18), which is equivalent to  $E_{\text{MIN}}$  in section 9.2.2.3, is evaluated using (12.19) with the conditional MAP reconstruction  $\mathbf{x}_{(i)}^{\text{MAP}}$  substituted for  $\mathbf{x}_{(i)}$ .

Joint conditional sampling of  $\mathbf{i}$  and  $\mathbf{x}$  could then be achieved using the method of *composition* (see e.g. [172]), which involves sampling  $\mathbf{i}$  from its reduced conditional  $p(\mathbf{i}|\boldsymbol{\omega}, \mathbf{y})$  (12.18) followed by sampling  $\mathbf{x}$  from its full conditional  $p(\mathbf{x}|\mathbf{i}, \boldsymbol{\omega}, \mathbf{y})$  equation (12.14). Equation (12.18) is a multivariate discrete distribution defined for all  $2^N$  possible permutations of  $\mathbf{i}$ . As the normalising constant  $c$  is unknown, direct sampling from this distribution will require  $2^N$  evaluations of (12.18). A suitable scheme is:

1. Evaluate (12.18) for all  $2^N$  possible permutations of  $\mathbf{i}$ .
2. Normalise sum of probabilities to unity and calculate cumulative distribution for  $\mathbf{i}$ .
3. Draw a random deviate  $u$  from a uniform distribution in the range  $0 < u < 1$ .
4. Select the value of  $\mathbf{i}$  whose cumulative probability is closest to  $u$  on the positive side (i.e. greater than  $u$ ).

Clearly such a sampling approach cannot be adopted in practice for any useful value of  $N$ , so some other scheme must be adopted. As stated earlier, MCMC methods allow a great deal of flexibility in the details of implementation. Issues such as which subsets of the parameters to sample and the precise order and frequency of sampling are left to the designer. These factors, while not affecting the asymptotic convergence of the sampler, will have great impact on short term convergence properties.

There is less computational burden in the full multidimensional sampling operation for  $\mathbf{x}_{(i)}$  (12.14), which is dominated by  $l$ -dimensional matrix-vector operations, so this can be performed on an occasional basis for the

whole block, in addition to the sub-block sampling described in a later section. This should eliminate any convergence problems which may arise as a result of posterior correlation between the  $x_t$ 's for a given  $\mathbf{i}$ .

### 12.5.3 Detection in sub-blocks

There are many possible adaptations to the direct multivariate sampling of (12.18) which will introduce trade-offs between computational complexity per iteration and number of iterations before convergence. Here we adopt a Gibbs sampling approach which performs the sampling given by (12.16) in small sub-blocks of size  $q$ , conditional upon switch values  $i_t$  and reconstructed data  $x_t$  from outside each sub-block. A compromise can thus be achieved between the computational load of operations (12.18) and (12.14) and convergence properties. An alternative scheme has recently been proposed for general state-space models in [36] in which individual elements  $i_t$  are conditionally sampled directly from the reduced conditional, given by equation 12.17 in our case. This is an elegant approach, but quite complex to implement and we leave a comparative evaluation as future work.

If we denote a particular sub-block of the data with length  $q$  samples by  $\mathbf{x}_{(q)}$  and the corresponding sub-block of detection indicators by  $\mathbf{i}_{(q)}$  then the required joint conditional for  $\mathbf{x}_{(q)}$  and  $\mathbf{i}_{(q)}$  is factorised as before:

$$p(\mathbf{x}_{(q)}, \mathbf{i}_{(q)} | \mathbf{x}_{-(q)}, \mathbf{i}_{-(q)}, \boldsymbol{\omega}, \mathbf{y}) = p(\mathbf{x}_{(q)} | \mathbf{x}_{-(q)}, \mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) p(\mathbf{i}_{(q)} | \mathbf{x}_{-(q)}, \mathbf{i}_{-(q)}, \boldsymbol{\omega}, \mathbf{y}) \quad (12.20)$$

where  $\mathbf{x}_{-(q)}$  and  $\mathbf{i}_{-(q)}$  are vectors containing the  $N - q$  data points and switch values respectively which lie outside the sub-block. The joint sample  $\mathbf{x}_{(q)}, \mathbf{i}_{(q)}$  is then obtained by composition sampling for  $\mathbf{i}_{(q)}$  and  $\mathbf{x}_{(q)}$  as described above.

The required full and reduced conditional distributions for  $\mathbf{x}_{(q)}$  and  $\mathbf{i}_{(q)}$  are obtained as an adaptation to the results for the complete data block (equations (12.18) and (12.14)). Since both distributions in (12.20) are conditioned upon surrounding reconstructed data points  $\mathbf{x}_{-(q)}$  it is equivalent for the purposes of conditional sampling to consider  $\mathbf{x}_{-(q)}$  as uncorrupted input data. In other words we can artificially set  $\mathbf{y}_{-(q)} = \mathbf{x}_{-(q)}$  and fix  $\mathbf{i}_{-(q)} = \mathbf{0}$  to obtain:

$$p(\mathbf{y} | \mathbf{i}_{(q)}, \mathbf{x}_{-(q)}, \mathbf{i}_{-(q)}, \boldsymbol{\omega}) = p(\mathbf{y}_{(q)}, \mathbf{y}_{-(q)} = \mathbf{x}_{-(q)} | \mathbf{i}_{(q)}, \mathbf{i}_{-(q)} = \mathbf{0}, \boldsymbol{\omega}) \quad (12.21)$$

and

$$p(\mathbf{x}_{(q)} | \mathbf{x}_{-(q)}, \mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) = p(\mathbf{x}_{(q)} | \mathbf{x}_{-(q)}, \mathbf{i}_{(q)}, \mathbf{i}_{-(q)} = \mathbf{0}, \boldsymbol{\omega}, \mathbf{y}_{(q)}, \mathbf{y}_{-(q)} = \mathbf{x}_{-(q)}) \quad (12.22)$$

This is a convenient form which allows direct use of equations (12.18) and (12.14) without any rearrangement. We now use the standard proportion-



ality between joint and conditional distributions (see appendix H.2):

$$p(\mathbf{i}_{(q)}|\mathbf{i}_{-(q)}, \boldsymbol{\omega}, \mathbf{y}) \propto p(\mathbf{i}|\boldsymbol{\omega}, \mathbf{y}) \quad (12.23)$$

$$p(\mathbf{x}_{(q)}|\mathbf{x}_{-(q)}, \mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) \propto p(\mathbf{x}|\mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) \quad (12.24)$$

to obtain the required results:

$$p(\mathbf{i}_{(q)}|\mathbf{x}_{-(q)}, \mathbf{i}_{-(q)}, \boldsymbol{\omega}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{i}, \boldsymbol{\omega})|_{(\mathbf{i}_{-(q)}=\mathbf{0}, \mathbf{y}_{-(q)}=\mathbf{x}_{-(q)})} p(\mathbf{i}) \quad (12.25)$$

and,

$$p(\mathbf{x}_{(q)}|\mathbf{x}_{-(q)}, \mathbf{i}, \boldsymbol{\omega}, \mathbf{y}) \propto p(\mathbf{x}|\mathbf{i}, \boldsymbol{\omega}, \mathbf{y})|_{(\mathbf{i}_{-(q)}=\mathbf{0}, \mathbf{y}_{-(q)}=\mathbf{x}_{-(q)})} \quad (12.26)$$

Thus equations (12.18) and (12.14) are used directly to perform sub-block sampling. The scheme is, then:

1. Sample the reduced conditional for  $\mathbf{i}_{(q)}$  using (12.25) and (12.18).  $\mathbf{i}_{(q)}$  is a binary vector of length  $q$  samples. Therefore  $2^q$  probability evaluations must be performed for each sub-block sample.
2. Sample the conditional for  $\mathbf{x}_{(q)}$  using (12.26) and (12.14) with the value of  $\mathbf{i}_{(q)}$  obtained in step 1. This involves drawing a random sample from the multivariate Gaussian defined in 12.14. Since  $\mathbf{i}_{-(q)}$  is constrained to be all zeros, the dimension of the Gaussian is equal to the number of non-zero elements in  $\mathbf{i}_{(q)}$  (a maximum of  $q$ ).

Note that in the important case  $q = 1$  these sampling steps are very efficient, requiring no matrix inversion/factorisation and only  $O(p)$  operations per sub-block sample.

#### 12.5.3.0.2 Sampling the Markov Chain transition probabilities

In the same way as the noise hyperparameters were treated as unknowns in the last section, the Markov chain transition probabilities  $P_{01}$  and  $P_{10}$  can be sampled as part of the detection procedure. The details, summarised in appendix H.3, are straightforward and involve sampling from beta densities which is a standard procedure [46]. We have observed that convergence of the algorithm is successful when the noise hyperparameters  $\alpha_n$  and  $\beta_n$  are known *a priori*. However, when processing real audio data and with these parameters treated as unknowns (as described in appendix H.3) performance was not good and the sampler chose very low values for  $P_{00} = 1 - P_{01}$  and high values for  $P_{11} = 1 - P_{10}$ . This is most likely a problem of modelling ambiguity: the sampler can either fit a very heavy-tailed noise model at most data points (setting  $i_t = 1$  nearly everywhere) or it can fit a high amplitude noise model at a few data points. It appears to choose the former case, which is not unreasonable since there will always be some low-level white noise at every data point in a real audio signal. We

are not attempting to model this low level noise component in this work, however (see [81, 72] for discussion of a related approach to this problem). In practice, then, we fix the transition probabilities to appropriate values for the type of impulsive noise present on a particular recording. We have found the methods to be fairly insensitive to the precise values of these parameters. For example, the values  $P_{00} = 0.93$  and  $P_{11} = 0.65$  have given successful results in all of the examples we have processed to date. A more elaborate approach (not pursued here) might attempt to estimate the transition probabilities from a ‘silent’ section of audio data in which no speech or music signal is present.

## 12.6 Results for EM and Gibbs sampler interpolation

Results are presented for a simple interpolation of a scratch-degraded audio signal, figure 12.1. The region of interpolation is indicated by the vertical dotted lines, which includes the scratch and a ‘guard zone’ of uncorrupted samples either side.

Figure 12.2 shows the results of interpolations using the Gibbs-estimated posterior mean (over 50 iterations following convergence) and the EM-estimated posterior mode (after 10 iterations). Both iterative schemes were informally judged to have converged in less than 10 iterations for this example. Figure 12.2(a) shows the missing data interpolator [192, 146], in which the corrupted samples are discarded prior to estimation. This interpolator makes a reasonable estimate, but clearly will not fit an interpolate close to samples in the guard zone. The common variance model (figure 12.2(b)) does a better job in the samples preceding the scratch, but is more wayward afterwards. Clearly the variance term is being biased upwards by some very high scratch amplitudes. The independent variance model (figure 12.2(c)) is able to fit the data in the guard zone almost perfectly, while performing a good interpolation of the main scratch, and even removing some very low amplitude noise material following the scratch. Thus we can expect greater fidelity to the source material with this method, and this is borne out by a ‘brighter’ sound to material restored in this way. There is not much difference to be seen in performance between the Gibbs and EM interpolation methods.

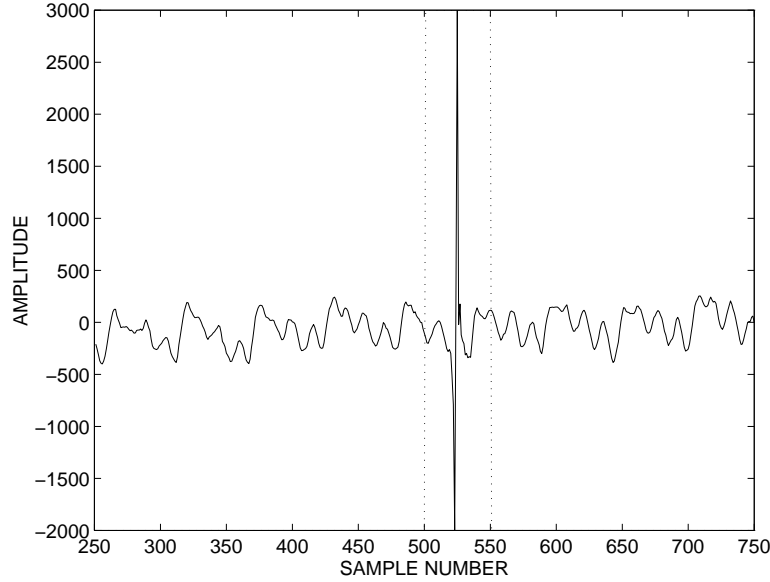


FIGURE 12.1. Corrupted audio waveform (78rpm disk)

## 12.7 Implementation of Gibbs sampler detection/interpolation

### 12.7.1 Sampling scheme

The Gibbs Sampler as implemented for joint detection and interpolation can be summarised as follows:

1. Assign initial values to unknowns:  $\mathbf{x}^0, \mathbf{i}^0, \boldsymbol{\omega}^0 = \{\mathbf{a}^0, \sigma_e^{2^0}, \sigma_{v_t}^{2^0} \mid t = 0 \dots N-1\}$ .
2. Repeat for  $i = 0 \dots N_{\max} - 1$ :
  - (a) Set  $\boldsymbol{\omega}^{i+1} = \boldsymbol{\omega}^i, \mathbf{i}^{i+1} = \mathbf{i}^i, \mathbf{x}^{i+1} = \mathbf{x}^i$ .
  - (b) Draw samples for unknown parameters:
    - i.  $\mathbf{a}^{i+1} \sim p(\mathbf{a} | \mathbf{i}^{i+1}, \mathbf{x}^{i+1}, \boldsymbol{\omega}_{-(\mathbf{a})}^{i+1}, \mathbf{y})$  (see (12.11))
    - ii.  $(\sigma_e^2)^{i+1} \sim p(\sigma_e^2 | \mathbf{i}^{i+1}, \mathbf{x}^{i+1}, \boldsymbol{\omega}_{-(\sigma_e^2)}^{i+1}, \mathbf{y})$  (see (12.12))
    - iii.  $(\sigma_{v_t}^2)^{i+1} \sim p(\sigma_{v_t}^2 | \mathbf{i}^{i+1}, \mathbf{x}^{i+1}, \boldsymbol{\omega}_{-(\sigma_{v_t}^2)}^{i+1}, \mathbf{y}), (t = 0 \dots N-1)$  (see (12.13))
  - (c) Draw samples from reconstructed data and switch values:
    - i. For all (non-overlapping) sub-blocks  $\mathbf{x}_{(q)}$  of length  $q$  samples in  $\mathbf{x}$  and  $\mathbf{i}_{(q)}$  in  $\mathbf{i}$ :
      - A.  $\mathbf{i}_{(q)}^{i+1} \sim p(\mathbf{i}_{(q)} | \mathbf{x}_{-(q)}^{i+1}, \mathbf{i}_{-(q)}^{i+1}, \boldsymbol{\omega}^{i+1}, \mathbf{y})$  (see (12.25))

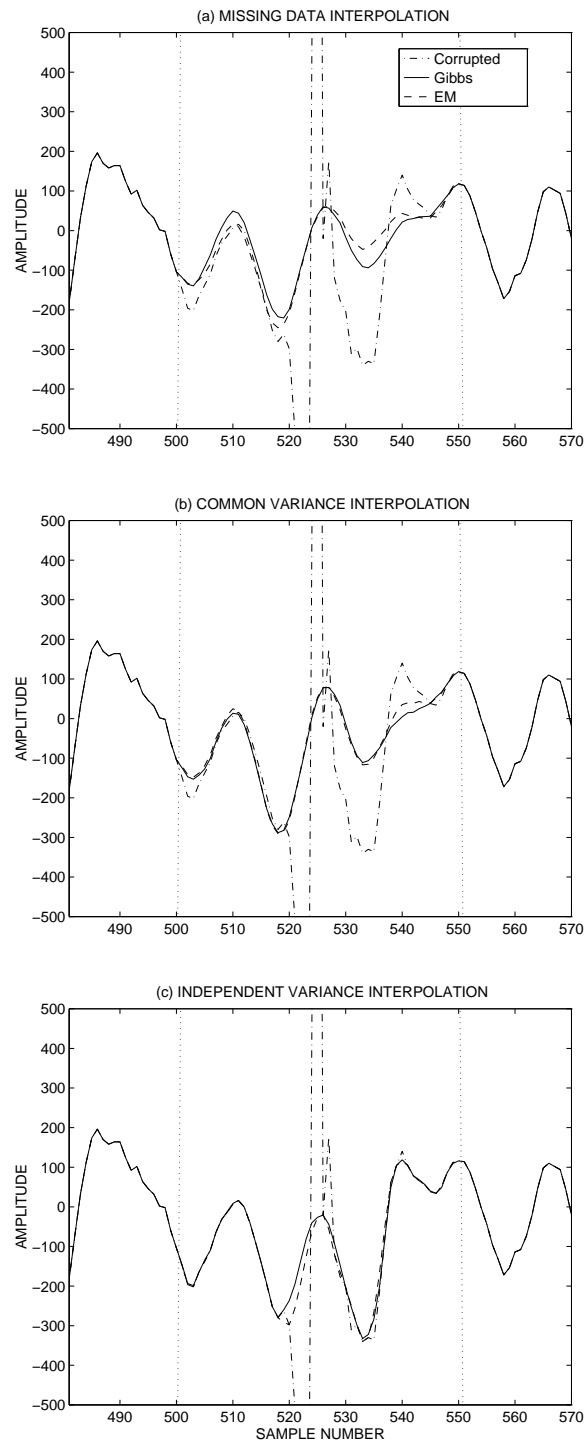


FIGURE 12.2. Interpolations using various noise models and AR(40)

- B.  $\mathbf{x}_{(q)}^{i+1} \sim p(\mathbf{x}_{(q)} | \mathbf{x}_{-(q)}^{i+1}, \mathbf{i}^{i+1}, \boldsymbol{\omega}^{i+1}, \mathbf{y})$  (see (12.26))
- ii. If  $((i \bmod M)=0)$ :  $\mathbf{x}^{i+1} \sim p(\mathbf{x} | \mathbf{i}^{i+1}, \boldsymbol{\omega}^{i+1}, \mathbf{y})$  (see (12.14))
- (d) (*Optional*) Sample top level hyperparameters:  $\alpha_v, \beta_v, P_{01}, P_{10}$  (see appendix H.3).
3. Calculate histograms and Monte Carlo estimates of posterior functionals (see (4.62)). For example, a Monte Carlo estimate of the posterior mean is given by:

$$E[\mathbf{x} | \mathbf{y}] \approx \frac{\sum_{i=N_0+1}^{N_{\max}} \mathbf{x}^i}{N_{\max} - N_0}$$

where  $N_0$  is the number of iterations before convergence is achieved.

Note that the multivariate sampling operation of 2(c)ii is carried out here once every  $M$  iterations.

### 12.7.2 Computational requirements

MCMC methods are generally considered to be highly computationally intensive, with typical applications requiring perhaps multiple runs, each with hundreds of iterations. Such schemes are clearly impractical in many engineering applications, particularly where large amounts of data must be processed. Such computational extremes will not, however, be necessary in cases where only point estimates of the data are required rather than a full analysis of all the posterior distributions. In the examples considered here it will be seen that useful results can be achieved after very few iterations, particularly if robust starting points are chosen for critical parameters. Nevertheless, each iteration of the scheme described in section 12.7.1 is itself quite intensive. Taking the items in the same order as section 12.7.1 we have:

- **Assigning initial values:**  $\mathbf{x}^0, \mathbf{i}^0$  and  $\boldsymbol{\omega}^0$  (step 1). This step need not take any significant computation since the unknowns can be assigned arbitrary random starting values. However, as discussed in the next section, it is beneficial to convergence of the algorithm if a reasonable starting point can be obtained using some other technique. The computation involved will usually be much smaller than the subsequent iterations of the Gibbs Sampler.
- **Sampling the AR parameters:**  $\mathbf{a}$  (step 2(b)i). This step requires a sample from a  $P$ -dimensional Gaussian distribution, given in equation 12.11. The most significant computation here is a square-root factorisation of the covariance matrix, which can be achieved using for example the Cholesky Decomposition (complexity  $O(P^3)$ ). Details of techniques for sampling multivariate Gaussians can be found in, for example, [46] or [168, pages 207-8].

- **Noise variances:**  $\sigma_e^2$  (step 2(b)ii) and  $\sigma_{v_t}^2$  (step 2(b)iii). The conditionals for all the noise variance terms are of Inverted Gamma form (see equations 2(b)ii and 2(b)iii and appendix A.4). Sampling from the Inverted Gamma distribution is straightforward (see e.g. [168], pp. 137-8 and [153]) and requires very little computation compared to other parts of the algorithm.
- **Joint sampling of switch and data values** (step 2(c)i). This step, as detailed in section 12.5.2, requires firstly the evaluation of the reduced conditional (equation 12.25) for all  $2^q$  possible values of the current detection sub-block  $\mathbf{i}_{(q)}$ , where  $q$  is the size of sub-block. Each such evaluation requires the calculation of  $\mathbf{x}_{\mathbf{i}}^{\text{MAP}}$ , involving the solution of a  $l_q \times l_q$  matrix-vector equation ( $l_q$  is the number of corrupted samples indicated by a particular  $\mathbf{i}_{(q)}$ ). This can be solved using a Cholesky decomposition of  $\Phi$ , which then facilitates the sampling of  $\mathbf{x}_{(q)}$ . The detection step rapidly becomes infeasible for all but small values of  $q$ . However, the experimental results show that small values of  $q$  give acceptable performance. In particular the case  $q = 1$  leads to a very efficient scheme which requires only  $O(NP)$  calculations for one complete sweep of all the sub-blocks. Note that  $q = 1$  retains the important advantage of sampling jointly for data values  $x_t$  and switch values  $i_t$ .

Having sampled  $\mathbf{i}_{(q)}$  from its reduced conditional the joint sample is completed by sampling for  $\mathbf{x}_{(q)}$  from its full conditional (12.26). The significant calculations for this operation will already have been carried out while sampling  $\mathbf{i}_{(q)}$ , since both stages require  $\mathbf{x}_{(\mathbf{i})}^{\text{MAP}}$  and the Cholesky decomposition of the matrix  $\Phi$ .

The precise balance of computational requirements will depend on the choice of sub-block size  $q$ , which in turn has a strong impact on the number of iterations required for convergence. In the absence of concrete results on the rate of convergence for the scheme, it is probably best to tune the sub-block size  $q$  by trial and error to give acceptable performance for a given application.

- **Sampling of complete data vector (interpolation):**  $\mathbf{x}$  (step 2(c)ii). Joint sampling of the complete data vector  $\mathbf{x}$  conditional upon all of the other unknowns is achieved using equation 12.14. As for the AR parameters this involves sampling from a multivariate Gaussian distribution, this time of dimension  $l$ , where  $l$  is the number of impulse-corrupted data points indicated by  $\mathbf{i}$ . Direct sampling, requiring matrix Cholesky factorisation, can be very intensive ( $O(l^3)$ ) for large  $l$ . This is significantly reduced, however, by observing that matrix  $\Phi$  is quite sparse in structure, owing to the Markov property of the assumed AR model. This sparse structure can be explicitly accounted for in the calculation of the Cholesky Decomposition, which

leads to a dramatic reduction in the computation (see [168], p.137 for discussion of the closely related problem of missing data interpolation). Possibly the best and most flexible way to take advantage of this structure, however, is by use of the standard state-space form for an AR model (see e.g. [5], pp.16-17). This then allows the use of the Kalman filter-smoother [5] for sampling of the whole data vector. Recent methods for achieving this efficiently in general state-space models can be found in [35, 61, 42]. In fact we implemented [42] because of its speed and efficient memory usage. Computations are then reduced to approximately  $O(lP^2)$ .

#### 12.7.2.1 Comparison of computation with existing techniques

The computational requirements of the Gibbs Sampling method can be compared with the detection and interpolation techniques of Vaseghi and Rayner [84, 191]. Here we have processing steps which are closely related in functionality and computation.

The parameter estimation stage of [191] involves estimation of AR parameters for the corrupted data. If a covariance-based AR parameter estimation method [119] is used then the computations will be of a similar complexity to the AR parameter sampling step in the Gibbs Sampler (note, however, that an autocorrelation method of estimation [119] would make this step faster for the simple scheme). Furthermore, the detection step in [191] is quite closely related to the sampling of detection vector and data vector in the Gibbs Sampler. In particular the computations are very similar (both  $O(NP)$  operations for one complete sweep through the data) when a sub-block length of  $q = 1$  is chosen for the Gibbs Sampler (the relationship between simple detectors and Bayesian detection schemes is discussed further in [70], appendix H). Finally, a close relationship can be found between the computation of the Least Squares AR-based interpolator in [191] and the Bayesian interpolation scheme. The computational requirements here are almost identical when the Least Squares-based interpolation is performed jointly over all corrupted samples in the data block. The only item not present in the simple scheme but required for the Gibbs Sampler is the estimation of noise hyperparameters, including noise variances. These can be neglected as they take up a very small fraction of the overall computation in the Gibbs Sampler.

Overall, then, one iteration of the Gibbs Sampler has roughly the same complexity as one complete pass of the simple detector/interpolator, with some small overheads in setting up the equations and drawing random variates. The chief difference lies in the large number of iterations required for the Gibbs Sampler compared with perhaps less than five passes required for the simple scheme.

## 12.8 Results for Gibbs sampler detection/interpolation

### 12.8.1 Evaluation with synthetic data

In order to demonstrate the operation of the proposed methods and for some comparison with other related techniques under idealised conditions we consider firstly a synthetic AR sequence with  $N = 300$  data points, see figure 12.3. The poles of the AR process have been selected arbitrarily with order  $p = 6$  and the excitation is white Gaussian noise with variance unity. The noise source  $v_t$  is Gaussian with independent variance components generated by sampling from an  $\text{IG}(0.3, 40)$  process. The switching process  $i_t$  is a first order Markov chain with transition probabilities defined by  $P_{00} = 0.93$  and  $P_{11} = 0.65$ . The resulting corrupted waveform is shown in figure 12.4, with the highest amplitude impulses not shown on the graph.

The Gibbs sampler was run under two noise modelling assumptions. The first model is the independent variance case, using the variance sampling methods given in (12.13). This corresponds to the true noise generation process and so can be considered as a somewhat idealised scenario.  $P_{00}$  and  $P_{11}$  are fixed at their true values, while all other parameters are sampled as unknowns.  $\alpha_v$  and  $\beta_v$  are reparameterised as  $m_v = \beta_v/\alpha_v$  and  $\alpha_v$  to give a more physically meaningful model (see appendix H.3). The transformed variables are treated as *a priori* independent in the absence of further knowledge.  $\alpha_v$  treated as a discrete variable with a uniform prior, taking on values uniformly spread over the range 0.1 to 3, while  $m_v$  is assumed independent of  $\alpha_v$  and given a very vague gamma prior  $G(10^{-10}, 10^{-10})$ . The second noise model assumes Gaussian noise with constant (unknown) variance but is otherwise identical to the first. This corresponds to the impulse models investigated in [79] with the generalisation that noise variances and AR parameters are unknown. A prior  $\text{IG}(10^{-10}, 10^{-10})$  is used for the common noise variance term. This noise model can be expected to give poorer results than the first, since it does not reflect the true noise generation process.

Both models were run for 1000 iterations of the Gibbs Sampler following a burn-in period of 100 iterations. The burn-in period was chosen as the number of iterations required for convergence of the Markov chain, as judged by informal examination of the sampled reconstructions. This figure is intentionally over-generous and was obtained after running the sampler with many different realisations of the corrupted input data. The detection vector  $\mathbf{i}$  was initialised to all zeros, while other parameters were initialised randomly. A minimal sub-block size  $q = 1$  was found to be adequate for the joint sampling of the switch and reconstructed data values 2(c)i, while multivariate sampling of the data elements 2(c)ii was performed once per iteration in sub-blocks of 100 samples.



MMSE reconstruction estimates were obtained as the arithmetic mean of the sampled reconstructions  $\mathbf{x}^i$  following the burn-in period. For the independent variance model, results are shown in figure 12.5. As hoped, comparison with the true signal shows accurate reconstruction of the original data sequence. Where significant errors can be seen, as in the vicinity of sample number 100, the impulsive noise can be seen to take very high values. As a result, the algorithm must effectively treat the data as ‘missing’ at these points. The corresponding result from the common variance model can be seen in figure 12.6. While the large impulses have all been removed, many smaller scale disturbances are left untouched. As expected, the common variance model has not been able to adjust *locally* to the scale of the noise artefacts.

While the reconstructed data is strictly all that is required in a restoration or enhancement framework, there is much additional information which can be extracted from the parameter and noise detection samples. These will assist in the assessment of convergence of the Markov chain, allow data analysis and parameter estimation and may ultimately help in the development of improved methods.

Consider, for example, samples drawn from the AR parameters,  $\mathbf{a}^i$ . Pole positions calculated from these sampled values can be plotted as scatter diagrams, which gives a useful pictorial representation of the posterior pole position distribution. This is shown for both noise models in figures 12.9 and 12.10. The true poles are at radii 0.99, 0.9 and 0.85 with angles  $0.1\pi$ ,  $0.3\pi$  and  $0.7\pi$ , respectively. The scatter diagram for the independent variance model shows samples clustered tightly around the most resonant pole (radius 0.99) with more posterior uncertainty around the remaining poles. The common variance model, however, indicates pole positions well away from their true values, since small impulses are still present in the data. Pole positions calculated from the MMSE AR parameter estimates are also indicated on both plots.

The excitation variance samples for the independent variance noise model are displayed in figures 12.11 and 12.12. Plotted against iteration number (figure 12.11) the variance can be seen to converge rapidly to a value of around 1.4, which is quite close to the true value of unity, while the histogram of samples following burn-in indicates that the variance is fairly well-determined at this value.

An important diagnostic is the detection vector  $\mathbf{i}$ . The normalised histogram of  $\mathbf{i}^i$  samples gives an estimate of posterior detection probabilities at each data sample. Regions of impulses are clearly indicated in figure 12.7, and a procedure which flags an impulse for all samples with probability values greater than 0.5 yields an error rate of 0.2% for this data sequence. By comparison, the common variance model gave an error rate of 5% (figure 12.8). Error rates are measured as the percentage of samples mis-classified by the detector.

Finally, histograms for  $\alpha_v$  and  $m_v$  are shown in figures 12.13 and 12.14. The true values are 0.3 and 40/1.3 respectively, obtained from the IG(0.3, 40) process which generated the noise variances.  $\alpha_v$  is well-centred around its true value, which is typical of all the datasets processed.  $m_v$  exhibits a much wider spread, centred at a value much larger than its true value. This can probably be accounted for by the relatively small number of noise samples used in the sampling of  $m_v$ . In any case the precise value of  $m_v$  within roughly an order of magnitude does not seem to have great impact on reconstruction results, which is a desirable consequence of the hierarchical noise modelling structure used.

### 12.8.2 Evaluation with real data

For evaluation with real data the method was tested using data digitally sampled at 44.1kHz and 16-bit integer resolution from degraded gramophone recordings containing voice and music material. The autoregressive parameters are assumed fixed for time intervals up to 25ms; hence data block lengths  $N$  of around 1100 samples are used. Processing for listening purposes can then proceed sequentially through the data, block by block, re-initialising the sampler for each new block and storing the reconstructed output data sequentially in a new data file. Continuity between blocks is maintained by using an overlap of  $P$  samples from block to block and fixing the first  $P$  elements of restored data in a new data block to equal the last  $P$  restored elements from the previous block. This is achieved by fixing the first  $P$  elements of the switch vector  $\mathbf{i}$  in the new block equal to zero and the first  $P$  elements of the new data vector  $\mathbf{y}$  to equal the last  $P$  restored output data points from the most recent block. Then the detection algorithm is only operated upon elements  $P \dots N - 1$  of  $\mathbf{i}$ .

Figures 12.16-12.18 show results from running one instance of the Gibbs Sampler for a single block of classical music, digitised from a typical noisy 78rpm recording (figure 12.15 shows this data block). An AR model order  $P = 30$  was chosen, which is adequate for representation of moderately complex classical music extracts. As for the synthetic example the switch values  $i_t$  were all initialised to ‘off’, i.e. no impulses present, the AR parameters and excitation variance were initialised by maximum likelihood (ML) from the corrupted data, and the noise variances were sampled randomly from their prior distribution. An informative gamma prior  $G(2, 0.0004)$  was assigned to  $m_v$ , a distribution whose mean value of 5000 was roughly estimated as the mean noise variance obtained from processing earlier blocks of the same dataset. A uniform prior was assigned to  $\alpha_v$  over the range 0.1 to 3. The Markov chain probabilities were fixed at  $P_{00} = 0.93$  and  $P_{11} = 0.65$  which had been found to be reasonable in earlier data.

The sampler was run for  $N_{\max} = 1000$  iterations with a ‘burn-in’ period of  $N_0 = 100$  iterations. Figure 12.16 shows the MMSE estimate reconstruction and figure 12.17 shows the estimated detection probabilities.

While most samples appear to have a small non-zero posterior probability of being an outlier (since there must be some continuous measurement noise present in all samples), time indices corresponding to high probabilities can be identified clearly with ‘spikes’ in the input waveform, figure 12.15. The detection procedure thus appears to be working as we would expect for the real data. All major defects visible to the naked eye have been rejected by the algorithm. Histograms for noise variance parameters are given in figures 12.18 and 12.19.

Some qualitative assessment of various possible sampling schemes can be obtained by examining the evolution with time of  $\sigma_e^2$ , since this parameter will be affected by impulses at any position in the waveform. The first 20 iterations of  $\sigma_e^2$  are displayed in figure 12.20(a) and (b) under various sub-block sampling schemes and initialising  $\mathbf{i}$  to be all zeros. Under our proposed scheme elements of  $i_t$  and  $x_t$  are sampled *jointly* in sub-blocks of size  $q$ . This is contrasted with the independent sampling scheme used in [129], adapted here to use our noise model and applied also in sub-blocks. Figure 12.20(a) shows the comparison for the minimal sub-block size  $q = 1$ . The joint sampling scheme is seen to converge significantly faster than the independent scheme which did not in fact reach the ‘true’ value of  $\sigma_e^2$  for many hundreds of iterations. In figure 12.20(b) the sub-block size is 4. Once again convergence is significantly faster for the joint sampling method, although the independent method does this time converge successfully after approximately 10 iterations. Note that the initial rate of change in  $\sigma_e^2$  does not depend strongly on the block length chosen. This is a result of the additional reconstruction operation (12.14) which is performed each iteration in large sub-blocks and helps to reduce the dependency on  $q$ . Thus we recommend that a small value of  $q$  be used in a practical situation. The convergence time of the independent sampling scheme was also found to be far less reliable than the joint scheme since it can take many iterations before very large impulses are first detected under the independent scheme. The differences in convergence demonstrated here will be a significant factor in typical applications such as speech and audio processing where speed is of the essence.

### 12.8.3 Robust initialisation

Initialisation of the sampler can be achieved in many ways but, if the method is to be of practical value, we should choose starting points that lead to very rapid and reliable convergence. Possibly the most critical initialisation is for the detection vector  $\mathbf{i}$ . Results presented thus far used a completely ‘blind’ initialisation with all elements of  $\mathbf{i}^0$  set to zero. Such a scheme shows the power of the sampling algorithm acting in ‘stand-alone’ mode. An alternative scheme might initialise  $\mathbf{i}^0$  to a robust estimate obtained from some other simple detection procedure. A rough and ready initial value for  $\mathbf{i}$  is obtained by thresholding the estimated AR excitation

sequence corresponding to the corrupted data and initial ML parameter estimates [191, 84], with a threshold set low enough to detect all sizeable impulses. One can then perform a least squares AR interpolation of the corrupted samples as in [191, 84] and then estimate  $\sigma_e^{2^0}$  and  $\mathbf{a}^0$  by standard maximum likelihood from the interpolated sequence. This gives the algorithm a good starting point from which the detection vector  $\mathbf{i}$  will converge very rapidly. Such an initialisation was used for processing of long sections of data where fast operation is essential. Convergence times were informally observed to be significantly reduced, with usable results obtained within the first 5-10 iterations. Take for example a synthetic example generated with the same modelling parameters as the example of section 12.8.1. We compare convergence properties for the proposed robust initialisation and for a random initialisation (parameters generated at random,  $\mathbf{x}^0$  set equal to  $\mathbf{y}$ ). Figure 12.21 shows the evolution with time of the innovation variance  $\sigma_e^2$ . This is plotted on a log-scale to capture the wide variations observed in the randomly initialised case. Note that in the random case  $\sigma_e^2$  was initialised as a uniform deviate between 0 and 100, so the huge values plotted are not just the result of a very unfortunate initialisation for this parameter alone. It is clear that from the robust starting point the sampler settles down at least twice as quickly as for random initialisation. Note also that the actual squared error between the long-term MMSE estimates from the samplers and the true data (available for this synthetic example) settled down to around 0.66 per sample in both cases, indicating that both initialisations give similar performance in the long term.

#### 12.8.4 Processing for audio evaluation

From a computational point of view it will not be possible to run the sampler for very many iterations if the scheme is to be of any practical use with long sections of data. Longer extracts processed using the sampler were thus limited to 15 iterations per data block. Restorations were taken as either the mean of the last 5 reconstructions or the last sampled reconstruction from the chain. This latter approach is appropriate for listening purposes since it can be regarded as a *typical* realisation of the restored data (see [161, 146, 168] for discussion of this point). This is an interesting matter, since most methods aim to find some estimator which minimises a mathematical *risk* function, such as a MAP or MMSE estimator. Here, however, we recognise that such estimators can sometimes lead to results which are *atypical* of the signals concerned. In short, they have to make a conservative choice of reconstruction in order to minimise the expected risk associated with the estimator. In audio processing this often manifests itself in reconstructions which are *oversmooth* compared to typical signals. A random sample drawn from the posterior distribution on the other hand will have a lower posterior probability than a MAP estimator but exhibits

characteristics more typical of the signal under consideration and therefore preferable for listening purposes. Informal evaluation in fact shows that either the sampled scheme or the MMSE scheme leads perceptually to very high quality restorations with the models used here. The criteria for evaluation are the amount of reduction in audible clicks/crackles, etc. and the degree of audible distortion (if any) to the program material itself compared with the corrupted input. The processed material is rendered almost entirely free from audible clicks and crackles in one single procedure, with minimal distortion of the underlying audio signal quality. The same degree of click reduction is not achievable by any other single procedure known to us for the removal of impulses from audio signals, and certainly not without much greater distortion of the sound quality.

### *12.8.5 Discussion of MCMC applied to audio*

Markov chain Monte Carlo methods allow for inference about very sophisticated probabilistic models which cannot easily be addressed using deterministic methods. Their use here allows for joint parameter estimation and reconstruction for signals in the presence of non-Gaussian switched noise processes, which are encountered in many important applications. The algorithms developed are computationally intensive, but give high quality results which we believe are not attainable using standard techniques. In particular, compared with earlier methods such as [191, 79], these techniques can reliably detect and remove impulses occurring on many different amplitude scales in one single procedure. We have tested the methods using synthetic data and examples obtained from corrupted voice and music recordings. It is hoped, however, that the methods are sufficiently general and robust to find application in a wide range of other areas, from communication systems to statistical data processing.

In this chapter we have provided only one example of the application of MCMC to audio signals. The methods can however be applied to any of the other applications described in the book. In particular, it is relatively straightforward to extend the work of this chapter to performing noise reduction as well as click removal, and to extend the signal models to ARMA rather than just AR. These areas are addressed in some of our recent papers [72, 73, 81]. For further applications in image enhancement, non-linear system modelling and Hidden Markov models, see [74, 105, 75, 106, 177, 178, 179, 182, 181, 180, 8, 50, 49].

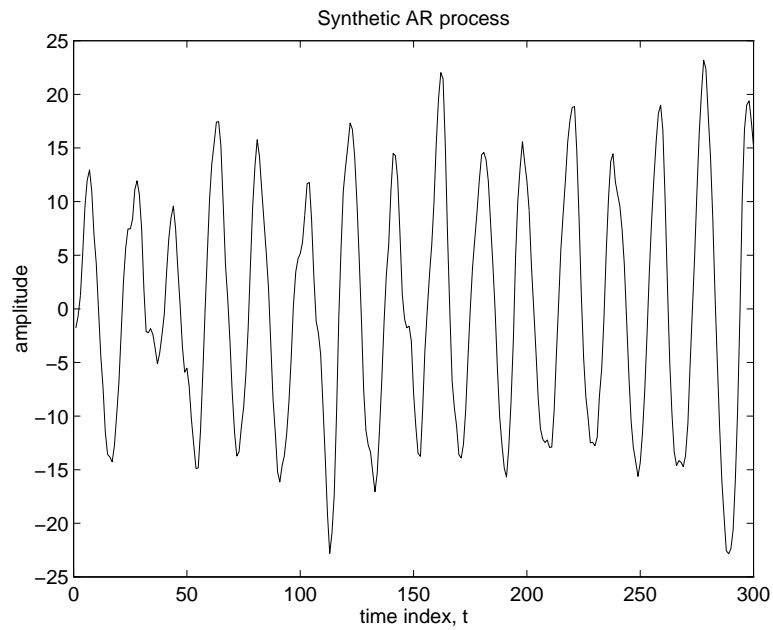


FIGURE 12.3. Synthetic AR data ( $p = 6$ )

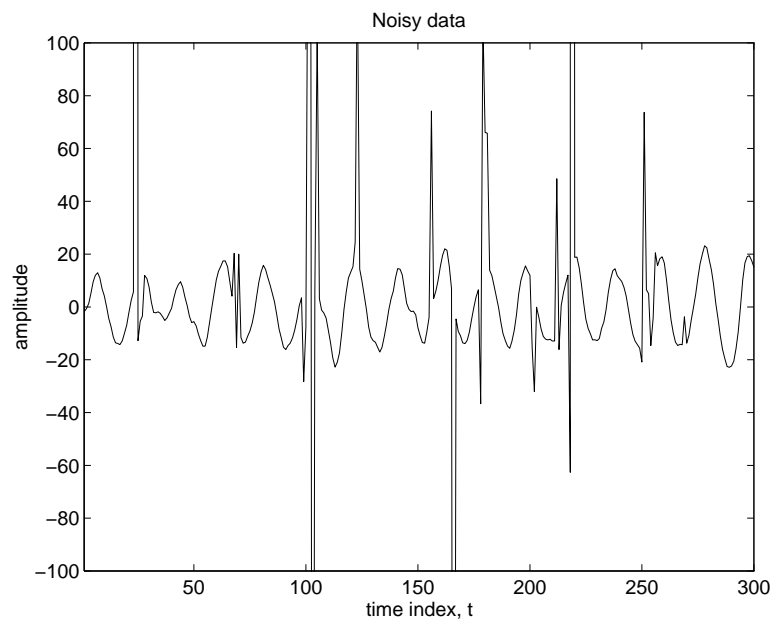


FIGURE 12.4. Noisy AR data

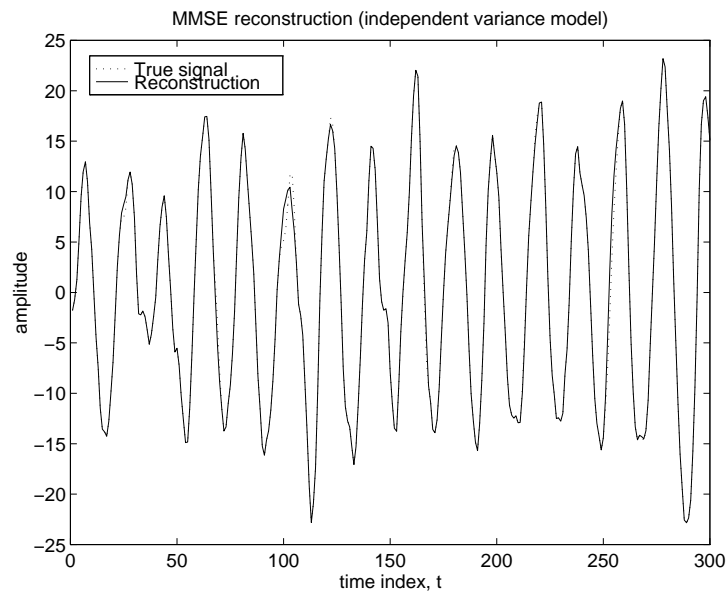


FIGURE 12.5. Reconstructed data (independent variance noise model)

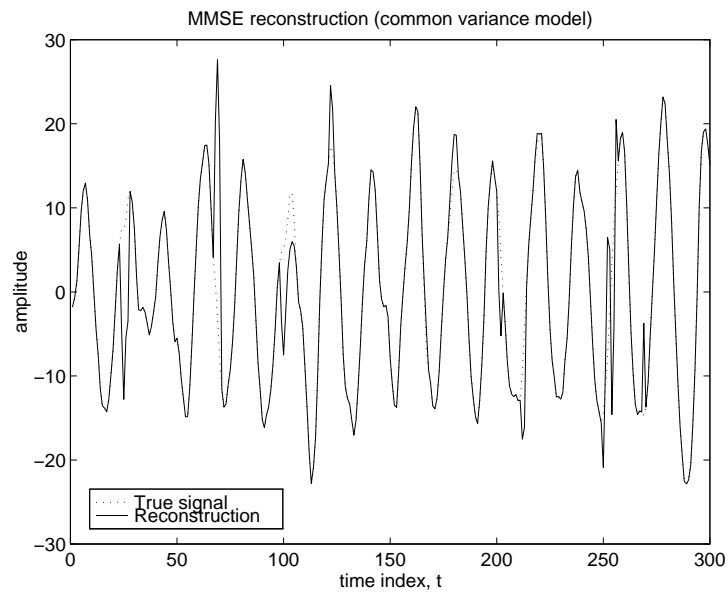


FIGURE 12.6. Reconstructed data (common variance noise model)

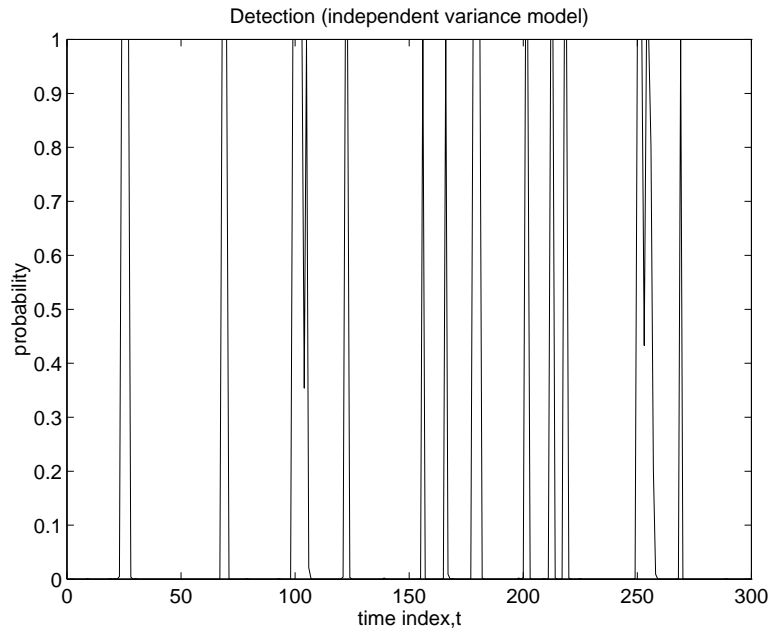


FIGURE 12.7. Detection probabilities (independent variance noise model)

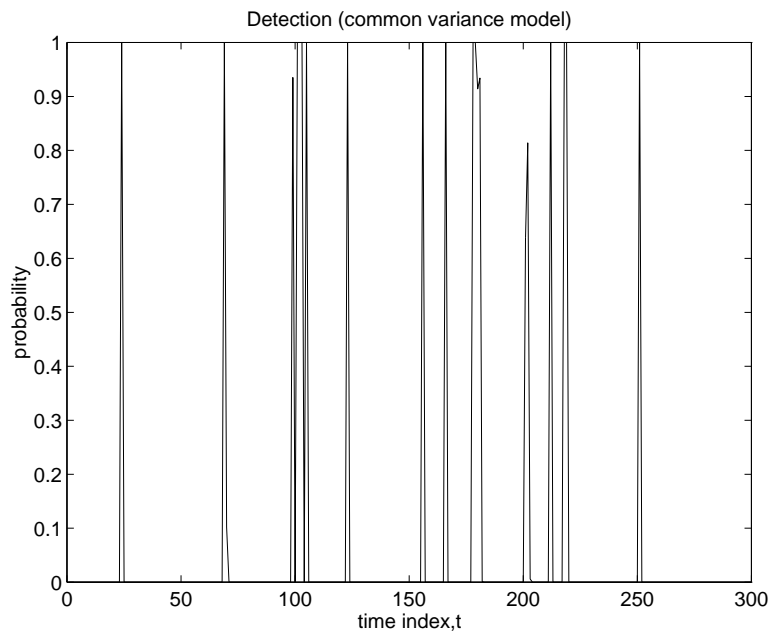


FIGURE 12.8. Detection probabilities (common variance noise model)



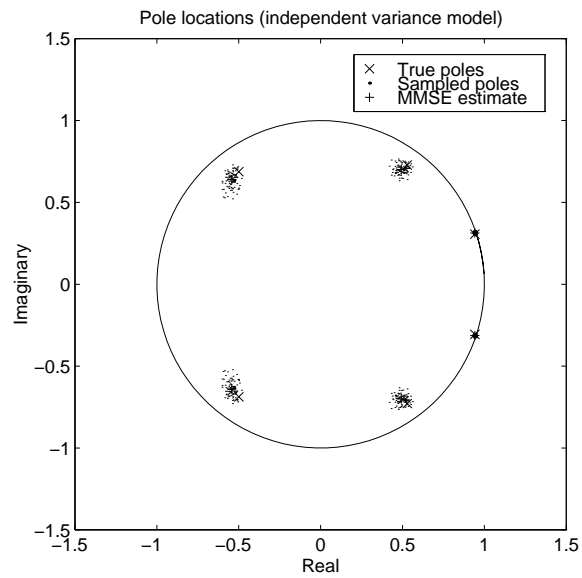


FIGURE 12.9. Reconstructed AR poles (independent variance noise model)

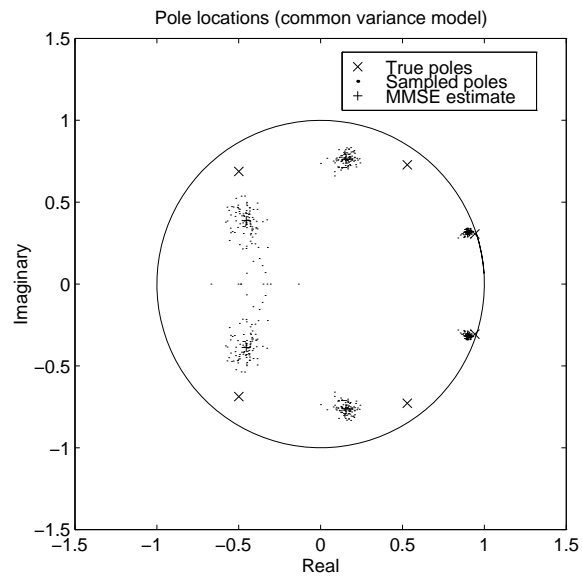


FIGURE 12.10. Reconstructed AR poles (common variance noise model)

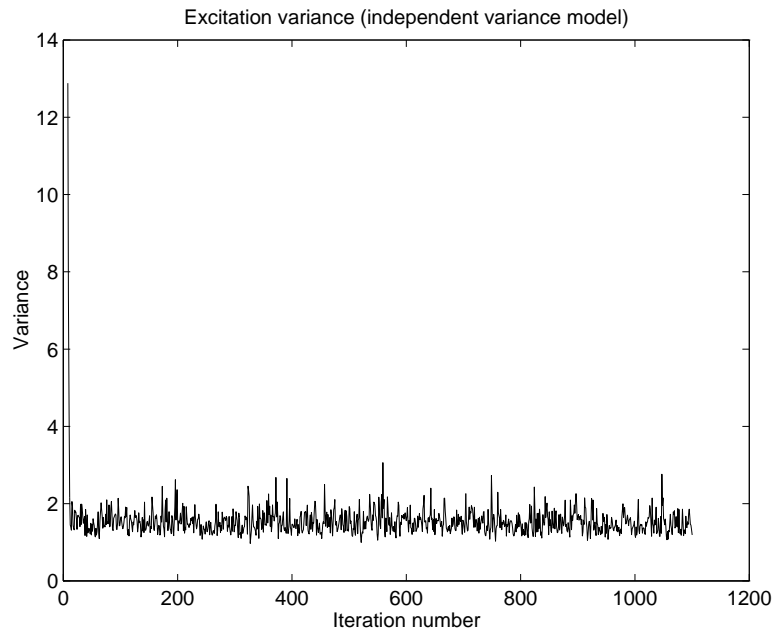


FIGURE 12.11. Excitation variance samples (independent variance noise model)

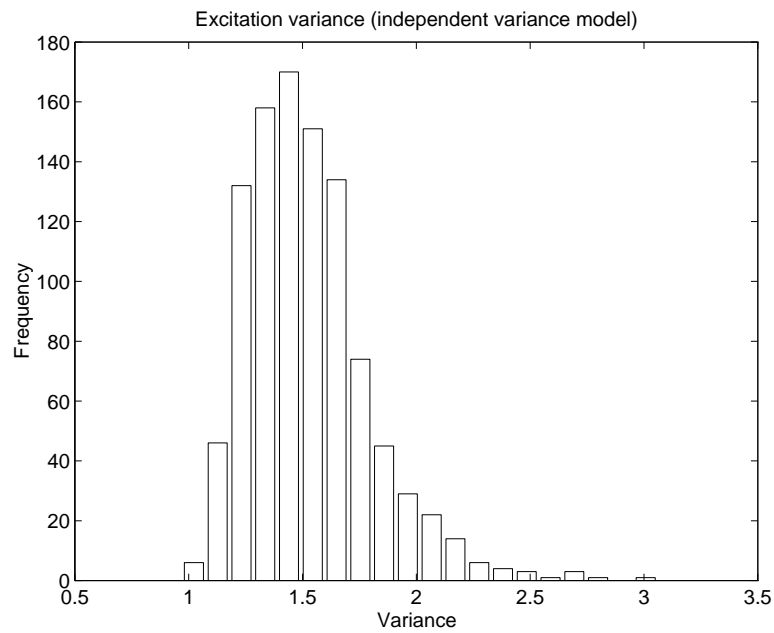
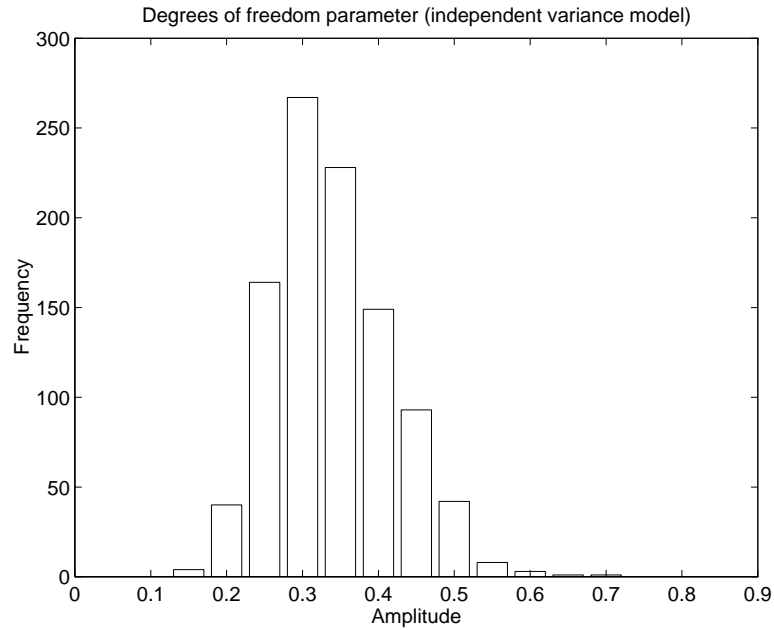
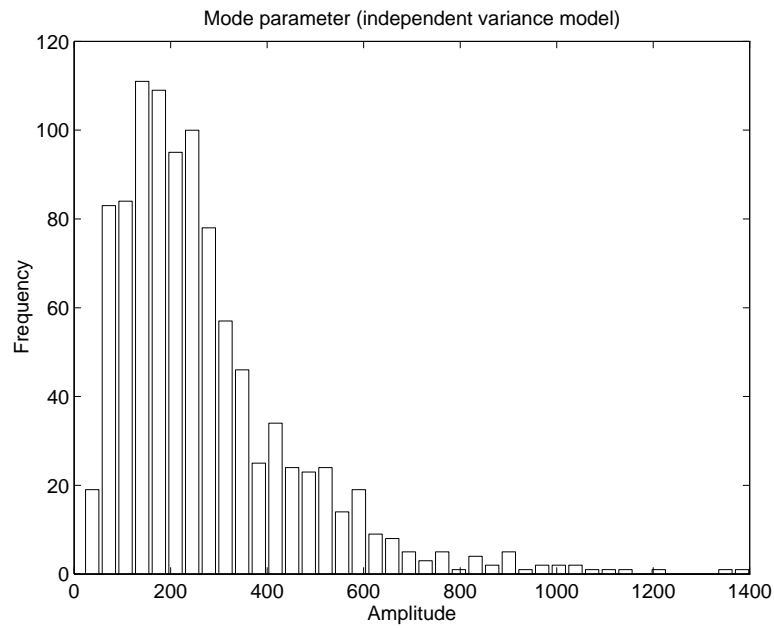


FIGURE 12.12. Excitation variance histogram (independent variance noise model)

FIGURE 12.13. Histogram of  $\alpha_v$  samples (independent variance noise model)FIGURE 12.14. Histogram of  $m_v$  samples (independent variance noise model)

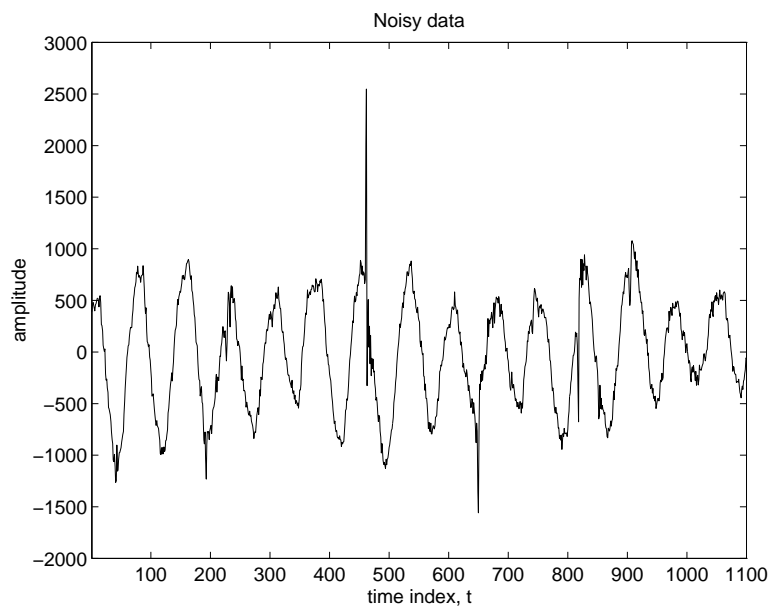


FIGURE 12.15. Noisy audio data

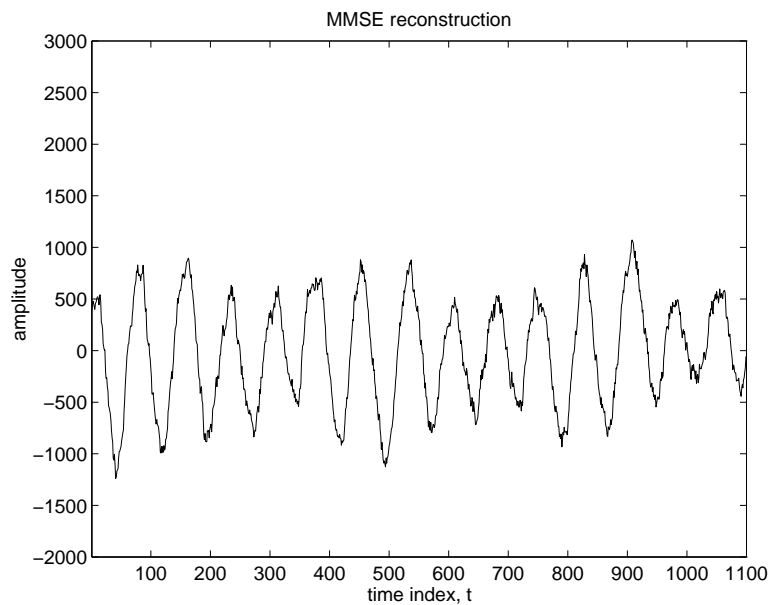


FIGURE 12.16. Reconstructed audio data

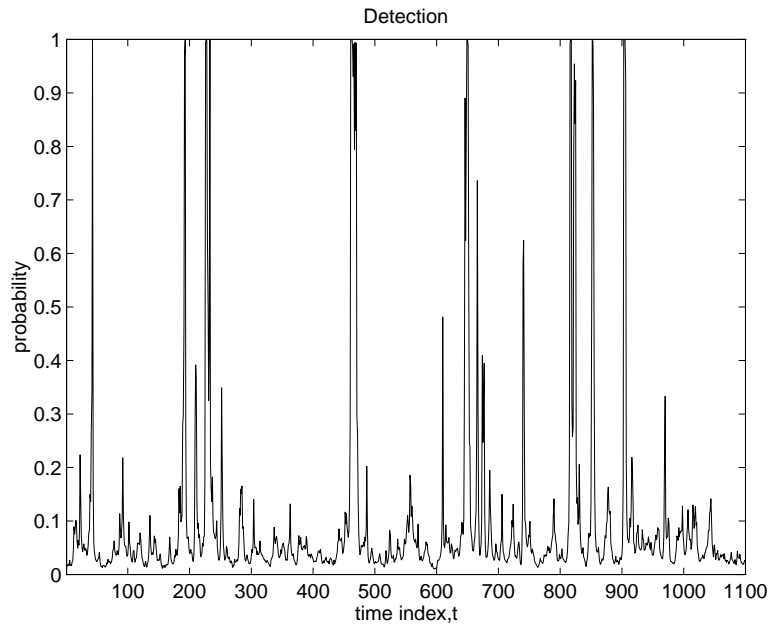


FIGURE 12.17. Estimated detection probabilities (real data)

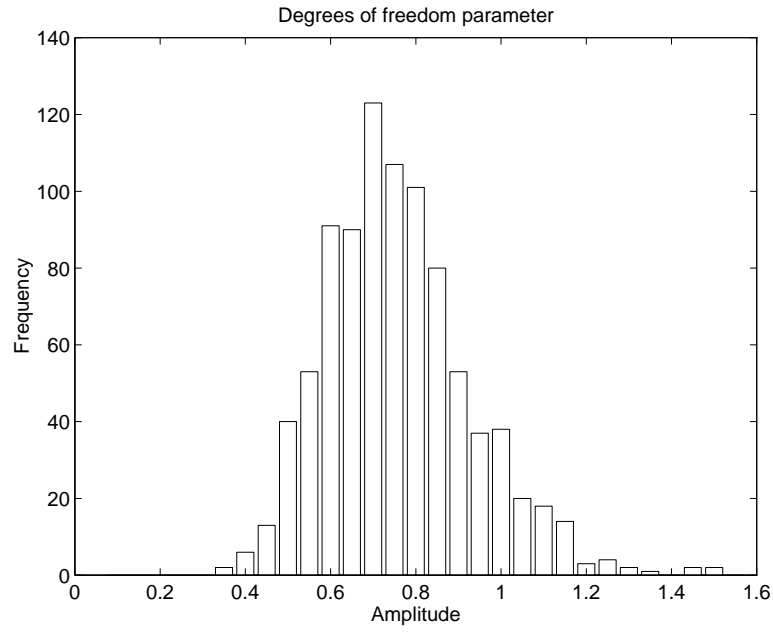


FIGURE 12.18. Histogram of  $\alpha_v$  samples (real data)

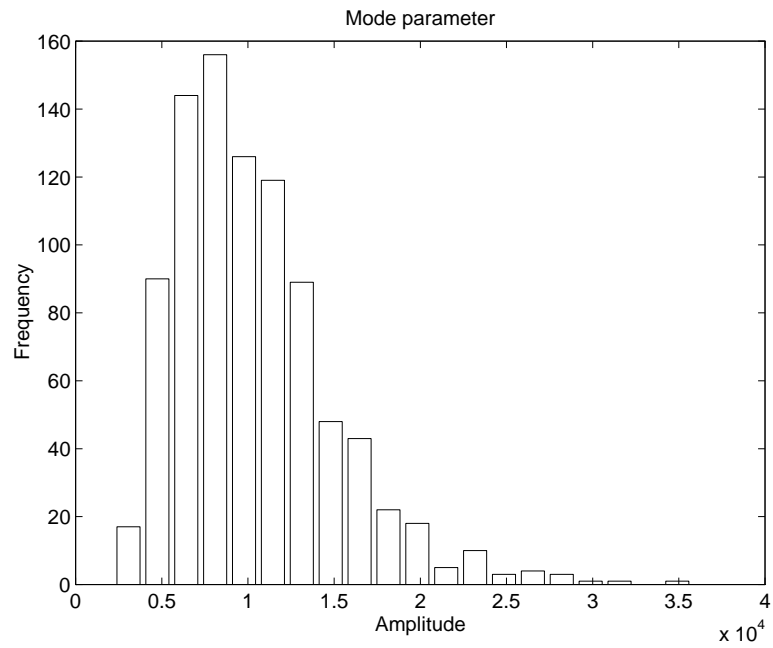


FIGURE 12.19. Histogram of  $m_v$  samples (real data)

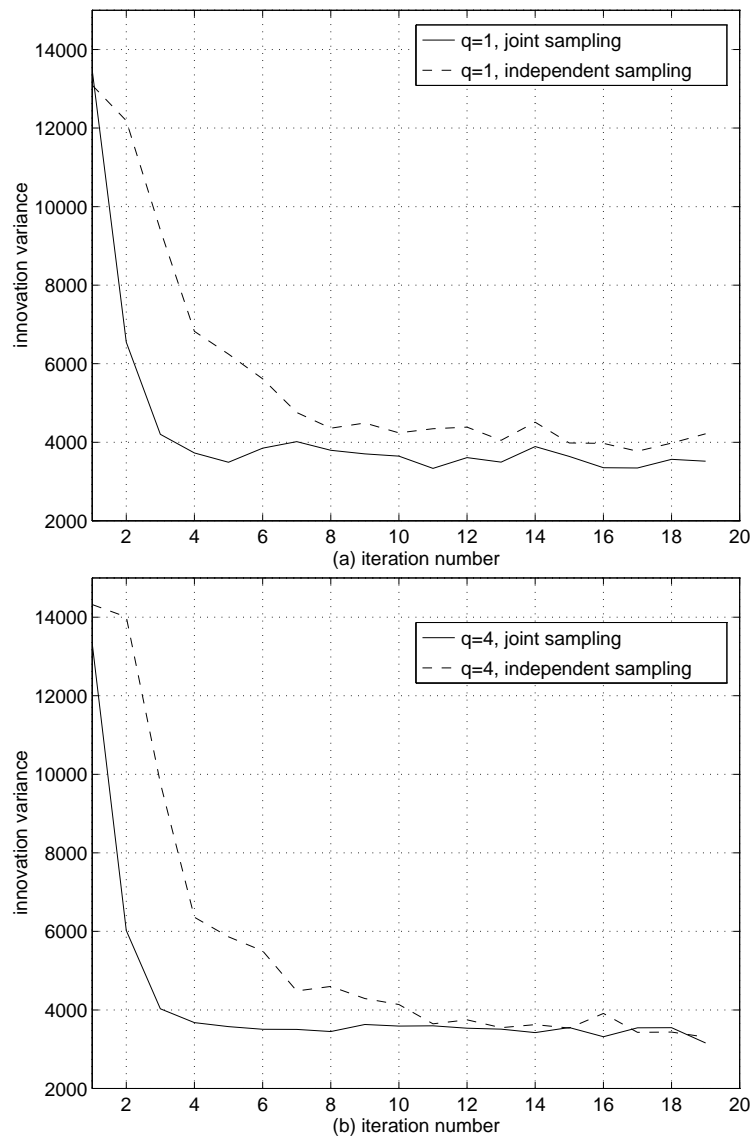


FIGURE 12.20. Convergence of excitation variance under different sampling strategies.

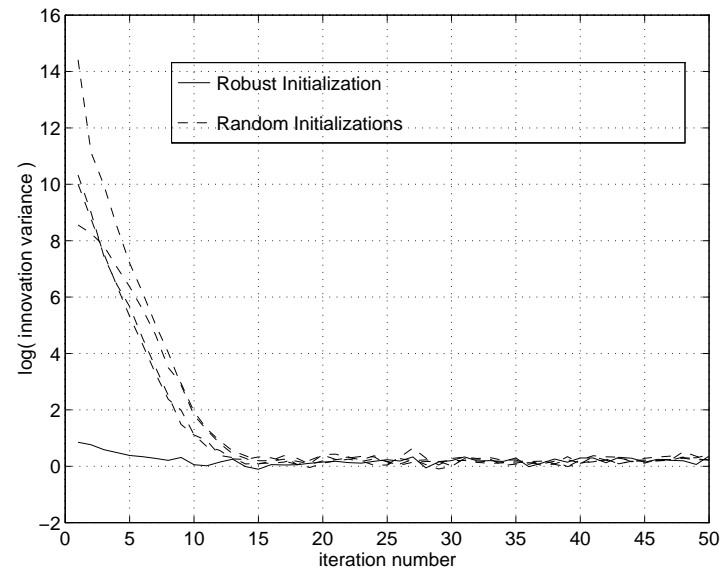


FIGURE 12.21. Comparison of robust and random initialisations



# 13

## Summary and Future Research Directions

In this book a wide range of techniques has been presented for the reduction of degradation in audio recordings. In Part II an attempt was made to provide a comprehensive overview of existing methods in click and hiss reduction, while also presenting some new developments and interpretations of standard techniques. The emphasis in the section on click removal was on model-based methods, which provide the opportunity to incorporate signal-specific prior information about audio. This is believed to be one of the key ways to improve on current technology and is the approach adopted throughout Part III on advanced methods. The section on background noise reduction, however, concentrates on spectral domain methods which can be regarded as non-parametric; this is a result of the history of the subject in which spectral domain methods have been applied almost without exception. Nevertheless, it is believed that model based methods will form a part of the future for hiss reduction. The difficulty here is that hiss reduction, being a global degradation, is a much more subtle and sensitive procedure than click removal, and model choice can be a critical consideration. Experimentation with standard speech based time series models such as the AR and ARMA models has led to some promising results for hiss reduction and joint hiss/click reduction, especially when the fully Bayesian methods of chapter 12 are employed [81, 72, 73], but it seems that to exceed the performance of the best spectral domain methods more realistic models will have to be used. These should include elements of non-stationarity, non-Gaussianity and the ability to model musical transients (note ‘attacks’, decays, performance noises) as well as steady state tonal sections. Combinations of elementary signals, such as wavelets or sinusoids,

with stochastic models (AR, ARMA, etc.) may go some way to providing the solutions and have already proved quite successful in high level modelling for coding, pitch transcription and synthesis [158, 169, 165, 194].

In Part III we present research work carried out over the last ten years into various aspects of digital audio restoration. The approach is generally model based and Bayesian, ranging from *empirical* Bayes methods which rely upon sub-optimal prior parameter estimation for efficiency (chapters 7-11) through to highly computationally intensive fully Bayesian methods which apply Monte Carlo simulation techniques to the optimal extraction of signals from noise (chapter 12). We present novel solutions to the areas of automatic pitch defect correction, an area where we are not aware of any other work, and for correction of low frequency noise pulses caused by breakages and deep scratches on gramophone discs and optical sound tracks. Finally we present a range of Bayesian techniques for handling click and crackle removal in a very accurate way.

### 13.1 Future directions and new areas

We have already stated that realistic signal modelling and sophisticated estimation procedures are likely to form the basis of future advances in audio restoration. A further area which has not been mentioned in depth is the incorporation of human perceptual criteria [137, 24] into the restoration process. It is well known that standard optimality criteria such as the mean-squared error (MSE) or MAP criterion are not tuned optimally to the human auditory system. Attempts have been made to incorporate some of the temporal and frequency domain (simultaneous) masking properties into coding and noise reduction systems, see chapter 6. However, these use heuristic arguments for incorporation of such properties. We believe that it may be possible to formulate a scheme using perceptually-based Bayesian cost functions, see chapter 4, combined with a Monte Carlo estimation method. This would formalise the approach and should show the potential of perceptual methods, which may as yet not have been exploited to the full. This is a topic of our current research.

We have covered in this text a very wide range of audio defects. One area which has scarcely been touched upon is that of non-linear distortion, a defect which is present in many recordings. This might be caused by saturation of electronics or magnetic recording media, groove deformation and tracing distortion in gramophone recordings or non-linearity in an analogue transmission path, which gives rise to unpleasant artefacts in the sound of the recording which should ideally be corrected. In general we will not have any very specific knowledge of the distortion mechanisms, so a fairly general modelling approach might be adopted. Many models are possible for non-linear time series, see e.g. [176]. In our initial work [130, 177, 179, 180]

we have adopted a cascade model in which the undistorted audio  $\{x_t\}$  is modelled as a linear autoregression and the non-linear distortion process as a ‘with memory’ polynomial non-linear autoregressive (NAR) filter containing terms up to a particular lag  $p$  and order  $q$ , so that the observed output  $y_t$  can be expressed as:

$$\begin{aligned}
 y_t = x_t + & \sum_{i_1=1}^p \sum_{i_2=1}^{i_1} b_{i_1 i_2} y_{t-i_1} y_{t-i_2} + \sum_{i_1=1}^p \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} b_{i_1 i_2 i_3} y_{t-i_1} y_{t-i_2} y_{t-i_3} \\
 & + \dots + \sum_{i_1=1}^p \sum_{i_2=1}^{i_1} \dots \sum_{i_q=1}^{i_{q-1}} b_{i_1 \dots i_q} y_{t-i_1} \dots y_{t-i_q}
 \end{aligned} \tag{13.1}$$

The model is illustrated in figure 13.1. The problem here is to identify the significant terms which are present in the non-linear expansion of equation 13.1 and to neglect insignificant terms, otherwise the number of terms in the model becomes prohibitively large. Results are very good for blind restoration of audio data which are artificially distorted with a NAR model (see e.g. figure 13.2). However, the models do not appear to be adequate for many of the distortions generally encountered in audio. The study of appropriate non-linear models would seem to be a fruitful area for future research. More specific forms of non-linear distortion are currently being studied, including clipping/pure saturation and coarsely quantised data. These are much more readily modelled and some progress is expected with such problems in the near future.

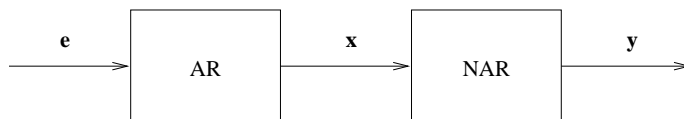


FIGURE 13.1. Block diagram of AR-NAR model

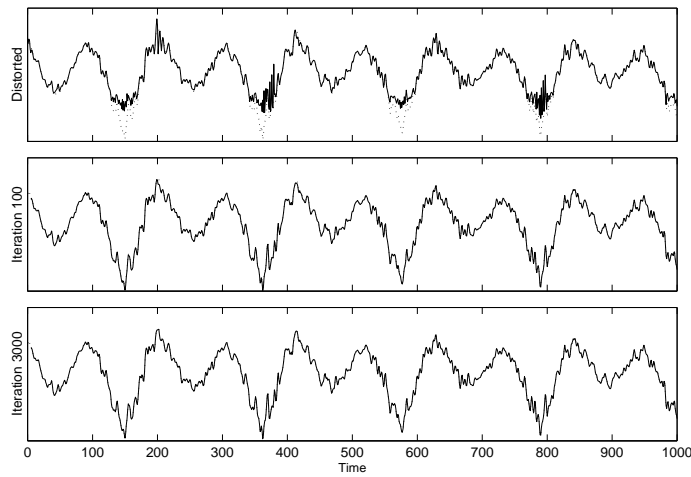


FIGURE 13.2. Example of AR-NAR restoration: Top - Non-linearly distorted input (solid), Undistorted input (dotted); Middle - Restored data (100 iterations of Gibbs sampler; Bottom - Restored data (3000 iterations of Gibbs sampler)

To conclude, we have presented here a set of algorithms which are based on principles which are largely new to audio processing. As such, this work might be regarded as a ‘new generation’ of audio restoration methods. The potential for improved performance has been demonstrated, although there is a corresponding increase in computational complexity. Computing power is still increasing rapidly, however, especially with the availability of high speed parallel DSP systems. In the same way that the first wave of restoration algorithms (see Part II) can now be implemented in real-time on standard DSP processors, it is hoped that the new generation of algorithms presented in Part III may soon be implemented in real-time on new processing platforms.

# Appendix A

## Probability Densities and Integrals

### A.1 Univariate Gaussian

The univariate Gaussian, or normal, density function with mean  $\mu$  and variance  $\sigma^2$  is defined for a real-valued random variable as:

$$\boxed{N(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}} \quad (\text{A.1})$$

*Univariate normal density*

### A.2 Multivariate Gaussian

The multivariate Gaussian probability density function (PDF) for a column vector  $\mathbf{x}$  with  $N$  real-valued components is expressed in terms of the mean vector  $\mathbf{m}_{\mathbf{x}}$  and the covariance matrix  $\mathbf{C}_{\mathbf{x}} = E[(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T]$  as:

$$\boxed{N_N(\mathbf{x}|\mathbf{m}, \mathbf{C}_{\mathbf{x}}) = \frac{1}{(2\pi)^{N/2} |\mathbf{C}_{\mathbf{x}}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T \mathbf{C}_{\mathbf{x}}^{-1} (\mathbf{x} - \mathbf{m}_{\mathbf{x}})\right)} \quad (\text{A.2})$$

*Multivariate Gaussian density*

An integral which is used on many occasions throughout the text is of the general form:

$$I = \int_{\mathbf{y}} \exp \left( -\frac{1}{2} (a + \mathbf{b}^T \mathbf{y} + \mathbf{y}^T \mathbf{C} \mathbf{y}) \right) d\mathbf{y} \quad (\text{A.3})$$

where  $d\mathbf{y}$  is interpreted as the infinitesimal volume element:

$$d\mathbf{y} = \prod_{i=1}^N dy_i$$

and the integral is over the real line in all dimensions, i.e. the single integration sign should be interpreted as:

$$\int_{\mathbf{y}} \equiv \int_{y_1=-\infty}^{\infty} \cdots \int_{y_N=-\infty}^{\infty}$$

For non-singular symmetric  $\mathbf{C}$  it is possible to form a ‘perfect square’ for the exponent and express  $I$  as:

$$I = \int_{\mathbf{y}} \exp \left( -\frac{1}{2} ((\mathbf{y} - \mathbf{m}_{\mathbf{y}})^T \mathbf{C} (\mathbf{y} - \mathbf{m}_{\mathbf{y}})) \right) \exp \left( -\frac{1}{2} \left( a - \frac{\mathbf{b}^T \mathbf{C}^{-1} \mathbf{b}}{4} \right) \right) d\mathbf{y} \quad (\text{A.4})$$

where

$$\mathbf{m}_{\mathbf{y}} = -\frac{\mathbf{C}^{-1} \mathbf{b}}{2}$$

Comparison with the multivariate PDF of A.2 which has unity volume leads directly to the result:

$$\boxed{\int_{\mathbf{y}} \exp \left( -\frac{1}{2} (a + \mathbf{b}^T \mathbf{y} + \mathbf{y}^T \mathbf{C} \mathbf{y}) \right) d\mathbf{y} = \frac{(2\pi)^{N/2}}{|\mathbf{C}|^{1/2}} \exp \left( -\frac{1}{2} \left( a - \frac{\mathbf{b}^T \mathbf{C}^{-1} \mathbf{b}}{4} \right) \right)} \quad (\text{A.5})$$

*Multivariate Gaussian integral*

This result can also be obtained directly by a transformation which diagonalises  $\mathbf{C}$  and this approach then verifies the normalisation constant given for the PDF of A.2.

### A.3 Gamma density

Another distribution which will be of use is the two parameter gamma density  $G(\alpha, \beta)$ , defined for  $\alpha > 0, \beta > 0$  as

$$G(y|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y) \quad (0 < y < \infty) \quad (\text{A.6})$$

*Gamma density*

$\Gamma()$  is the Gamma function (see e.g. [11]), defined for positive arguments. This distribution with its associated normalisation enables us to perform marginalisation of scale parameters with Gaussian likelihoods and a wide range of parameter priors (including uniform, Jeffreys, Gamma and Inverse Gamma (see [121]) priors) which all require the following result:

$$\int_{y=0}^{\infty} y^{\alpha-1} \exp(-\beta y) dy = \Gamma(\alpha)/\beta^\alpha \quad (\text{A.7})$$

*Gamma integral*

Furthermore the mean, mode and variance of such a distribution are obtained as:

$$\mu = E[Y] = \alpha/\beta \quad (\text{A.8})$$

$$m = \underset{y}{\operatorname{argmax}}(p(y)) = (\alpha - 1)/\beta \quad (\text{A.9})$$

$$\sigma^2 = E[(Y - \mu)^2] = \alpha/\beta^2 \quad (\text{A.10})$$

### A.4 Inverted Gamma distribution

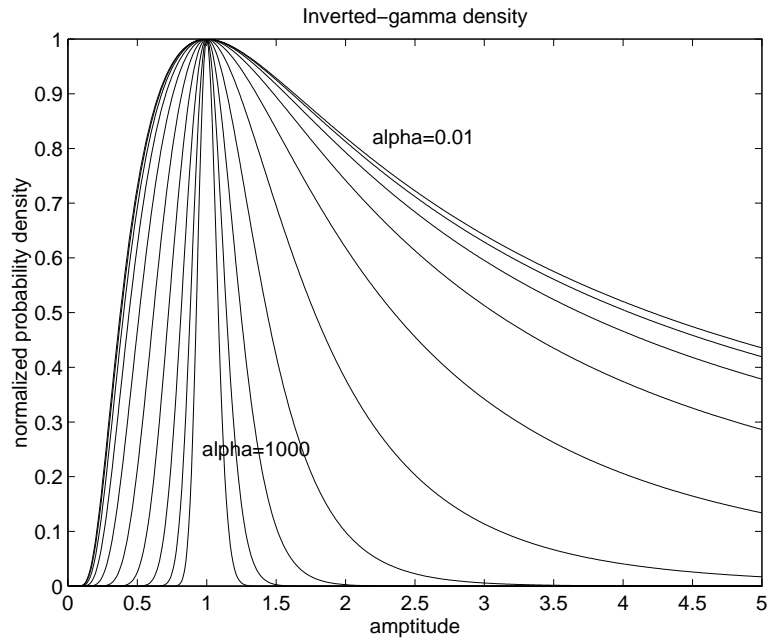
A closely related distribution is the inverted-gamma distribution,  $IG(\alpha, \beta)$  which describes the distribution of the variable  $1/Y$ , where  $Y$  is distributed as  $G(\alpha, \beta)$ :

$$IG(y|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{-(\alpha+1)} \exp(-\beta/y) \quad (0 < y < \infty) \quad (\text{A.11})$$

*Inverted-gamma density*

The IG distribution has a unique maximum at  $\beta/(\alpha + 1)$ , mean value  $\beta/(\alpha - 1)$  (for  $\alpha > 1$ ) and variance  $\beta^2/((\alpha - 1)^2(\alpha - 2))$  (for  $\alpha > 2$ ).

It is straightforward to see that the improper Jeffreys prior  $p(x) = 1/x$  is obtained in the limit as  $\alpha \rightarrow 0$  and  $\beta \rightarrow 0$ .

FIGURE A.1. Inverted-gamma family with mode=1,  $\alpha = 0.01 \dots 1000$ 

The family of IG distributions is plotted in figure A.1 as  $\alpha$  varies over the range 0.01 to 1000 and with maximum value fixed at unity. The variety of distributions available indicates that it is possible to incorporate either very vague or more specific prior information about variances by choice of the mode and degrees of freedom of the distribution. With high values of  $\alpha$  the prior is very tightly clustered around its mean value, indicating a high degree of prior belief in a small range of values, while for smaller  $\alpha$  the prior can be made very diffuse, tending in the limit to the uninformative Jeffreys prior. Values of  $\alpha$  and  $\beta$  might be chosen on the basis of mean and variance information about the unknown parameter or from estimated percentile positions on the axis.



# Appendix B

## Matrix Inverse Updating Results and Associated Properties

Here we are concerned with updates to a non-singular ( $N \times N$ ) matrix  $\Phi$  which are of the form:

$$\Phi \rightarrow \Phi + \mathbf{u} \mathbf{v}^T \quad (\text{B.1})$$

for some column vectors  $\mathbf{u}$  and  $\mathbf{v}$ . Results given here can be found in [153] (and many other texts) but since these results are used several times in the main text they are presented here in full for reference purposes.

Consider firstly a block partitioned matrix  $\mathbf{A}$ :

$$\mathbf{A} = \begin{bmatrix} \Phi & \mathbf{U} \\ \mathbf{V}^T & \mathbf{W} \end{bmatrix} \quad (\text{B.2})$$

Matrix  $\mathbf{A}$  can be inverted by row/column manipulations in two ways to give the following equivalent results:

$$\begin{aligned} & \mathbf{A}^{-1} \\ &= \begin{bmatrix} (\Phi - \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T)^{-1} & -(\Phi - \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T)^{-1}\mathbf{U}\mathbf{W}^{-1} \\ -\mathbf{W}^{-1}\mathbf{V}^T(\Phi - \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T)^{-1} & (\mathbf{W}^{-1} + \mathbf{W}^{-1}\mathbf{V}^T(\Phi - \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T)^{-1}\mathbf{U}\mathbf{W}^{-1}) \end{bmatrix} \end{aligned} \quad (\text{B.3})$$

$$\begin{aligned} &= \begin{bmatrix} (\Phi^{-1} + \Phi^{-1}\mathbf{U}(\mathbf{W} - \mathbf{V}^T\Phi^{-1}\mathbf{U})^{-1}\mathbf{V}^T\Phi^{-1}) & -\Phi^{-1}\mathbf{U}(\mathbf{W} - \mathbf{V}^T\Phi^{-1}\mathbf{U})^{-1} \\ -(\mathbf{W} - \mathbf{V}^T\Phi^{-1}\mathbf{U})^{-1}\mathbf{V}^T\Phi^{-1} & (\mathbf{W} - \mathbf{V}^T\Phi^{-1}\mathbf{U})^{-1} \end{bmatrix} \end{aligned} \quad (\text{B.4})$$

An intermediate stage of this inversion procedure obtains an upper or lower block triangular matrix form from which the determinant of  $\mathbf{A}$  can be obtained as

$$|\mathbf{A}| = |\mathbf{\Phi}| |(\mathbf{W} - \mathbf{V}^T \mathbf{\Phi}^{-1} \mathbf{U})| = |\mathbf{W}| |(\mathbf{\Phi} - \mathbf{U} \mathbf{W}^{-1} \mathbf{V}^T)| \quad (\text{B.5})$$

If we equate the top left hand elements of the two inverse matrices B.3 and B.4 the result is

$$(\mathbf{\Phi} - \mathbf{U} \mathbf{W}^{-1} \mathbf{V}^T)^{-1} = \mathbf{\Phi}^{-1} + \mathbf{\Phi}^{-1} \mathbf{U} (\mathbf{W} - \mathbf{V}^T \mathbf{\Phi}^{-1} \mathbf{U})^{-1} \mathbf{V}^T \mathbf{\Phi}^{-1} \quad (\text{B.6})$$

which is the well known Woodbury update formula or Matrix Inversion Lemma.

If  $\mathbf{U} = \mathbf{u}$ ,  $\mathbf{V} = \mathbf{v}$  and  $\mathbf{W} = -1$  the update is then as required for B.1, giving the Sherman-Morrison formula:

$$(\mathbf{\Phi} + \mathbf{u} \mathbf{v}^T)^{-1} = \mathbf{\Phi}^{-1} - \frac{\mathbf{\Phi}^{-1} \mathbf{u} \mathbf{v}^T \mathbf{\Phi}^{-1}}{1 + \mathbf{v}^T \mathbf{\Phi}^{-1} \mathbf{u}} \quad (\text{B.7})$$

which is an order  $\mathcal{O}(N^2)$  operation and hence more efficient than direct matrix inversion using Cholesky decomposition, an  $\mathcal{O}(N^3)$  operation.

Suppose we are in fact solving a set of linear equations at time index  $n$  such that  $\mathbf{\Phi}_n \mathbf{x}_n = \boldsymbol{\theta}_n$  and the current solution  $\mathbf{x}_n$  is known. Matrix  $\mathbf{\Phi}_n$  is now updated according to:

$$\mathbf{\Phi}_{n+1} = \mathbf{\Phi}_n + \mathbf{u}_n \mathbf{u}_n^T \quad (\text{B.8})$$

while  $\boldsymbol{\theta}_n$  is updated as

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \mathbf{u}_n d_n. \quad (\text{B.9})$$

Some further manipulation and the use of B.7 and B.5 leads to the following extended form of the recursive least squares (RLS) update employed in adaptive filtering:

$$\mathbf{k}_{n+1} = \frac{\mathbf{\Phi}_n^{-1} \mathbf{u}_n}{1 + \mathbf{u}_n^T \mathbf{\Phi}_n^{-1} \mathbf{u}_n} \quad (\text{B.10})$$

$$\alpha_{n+1} = d_n - \mathbf{x}_n^T \mathbf{u}_n \quad (\text{B.11})$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{k}_{n+1} \alpha_{n+1} \quad (\text{B.12})$$

$$\mathbf{\Phi}_{n+1}^{-1} = \mathbf{\Phi}_n^{-1} - \mathbf{k}_{n+1} \mathbf{u}_n^T \mathbf{\Phi}_n^{-1} \quad (\text{B.13})$$

$$|\mathbf{\Phi}_{n+1}| = |\mathbf{\Phi}_n| (1 + \mathbf{u}_n^T \mathbf{\Phi}_n^{-1} \mathbf{u}_n) \quad (\text{B.14})$$

$$E_{n+1}^{\text{MIN}} = E_n^{\text{MIN}} + \alpha_{n+1} (d_n - \mathbf{x}_{n+1}^T \mathbf{u}_n) \quad (\text{B.15})$$

where  $E_n^{\text{MIN}}$  is the minimum sum of error squares for the least squares problem solved by  $\mathbf{x}_n = \mathbf{\Phi}_n^{-1} \boldsymbol{\theta}_n$

# Appendix C

## Exact Likelihood for AR Process

In this appendix we derive the exact likelihood for a stable (stationary) AR process with parameters  $\{\mathbf{a}, \sigma_e^2\}$ . The conditional likelihood has already been obtained as:

$$p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}\right). \quad (\text{C.1})$$

In order to obtain the true likelihood for the whole data block  $\mathbf{x}$  use the probability chain rule:

$$p(\mathbf{x} | \mathbf{a}) = p(\{\mathbf{x}_0, \mathbf{x}_1\} | \mathbf{a}) = p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{a}) p(\mathbf{x}_0 | \mathbf{a}) \quad (\text{C.2})$$

and the missing term is thus  $p(\mathbf{x}_0, \mathbf{a})$ , the joint PDF for  $P$  samples of the AR process. Since the data is assumed zero mean and Gaussian we can write

$$p(\mathbf{x}_0 | \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{P/2} |\mathbf{M}_{\mathbf{x}_0}|^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}_0^T \mathbf{M}_{\mathbf{x}_0}^{-1} \mathbf{x}_0\right) \quad (\text{C.3})$$

where  $\mathbf{M}_{\mathbf{x}_0}$  is the covariance matrix for  $P$  samples of data with unit variance excitation, which is well-defined for a stable model.

The exact likelihood expression is thus:

$$p(\mathbf{x} | \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N}{2}} |\mathbf{M}_{\mathbf{x}_0}|^{1/2}} \exp\left(-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{M}_{\mathbf{x}}^{-1} \mathbf{x}\right) \quad (\text{C.4})$$

where

$$\mathbf{M}_{\mathbf{x}}^{-1} = \mathbf{A}^T \mathbf{A} + \begin{bmatrix} \mathbf{M}_{\mathbf{x}_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (\text{C.5})$$

is the inverse covariance matrix for a block of  $N$  samples with  $\sigma_e^2 = 1$ .

Matrix  $\mathbf{M}_{\mathbf{x}_0}$  can easily be calculated since it is composed of elements from the covariance sequence of the AR process which may be known beforehand (e.g. if an asymptotic approximation is made for the parameter estimation stage then the covariance values will be specified before estimation of the parameters).

In any case  $\mathbf{M}_{\mathbf{x}_0}^{-1}$  is easily obtained owing to the reversible character of the AR process which implies that  $\mathbf{M}_{\mathbf{x}}^{-1}$  must be *persymmetric*, i.e. symmetrical about both principal axes. Consider partitioning  $\mathbf{A}$  for  $N \geq 2P$  into three components:  $\mathbf{A}_0$  contains the first  $P$  columns of  $\mathbf{A}$ ,  $\mathbf{A}_{1b}$  contains the last  $P$  columns of  $\mathbf{A}$  and  $\mathbf{A}_{1a}$  contains the remaining columns, so that

$$\mathbf{A} = [\mathbf{A}_0 \ \mathbf{A}_{1a} \ \mathbf{A}_{1b}] \quad (\text{C.6})$$

and

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \mathbf{A}_0^T \mathbf{A}_0 & \mathbf{A}_0^T \mathbf{A}_{1a} & \mathbf{A}_0^T \mathbf{A}_{1b} \\ \mathbf{A}_{1a}^T \mathbf{A}_0 & \mathbf{A}_{1a}^T \mathbf{A}_{1a} & \mathbf{A}_{1a}^T \mathbf{A}_{1b} \\ \mathbf{A}_{1b}^T \mathbf{A}_0 & \mathbf{A}_{1b}^T \mathbf{A}_{1a} & \mathbf{A}_{1b}^T \mathbf{A}_{1b} \end{bmatrix} \quad (\text{C.7})$$

Comparing (C.7) with (C.5) and using the persymmetric property of  $\mathbf{M}_{\mathbf{x}}^{-1}$  gives the following expression for  $\mathbf{M}_{\mathbf{x}_0}^{-1}$ :

$$\mathbf{M}_{\mathbf{x}_0}^{-1} = (\mathbf{A}_{1b}^T \mathbf{A}_{1b})^R - \mathbf{A}_0^T \mathbf{A}_0 \quad (\text{C.8})$$

where the operator  $R$  reverses row and column ordering of a matrix. Since elements of  $\mathbf{A}_0$  and  $\mathbf{A}_{1b}$  are AR coefficient values elements of  $\mathbf{M}_{\mathbf{x}_0}^{-1}$  are quadratic in the AR coefficients. Since  $\mathbf{A}^T \mathbf{A}$  is also quadratic in the coefficients it follows that the exponent of the exact likelihood expression  $\mathbf{x}^T \mathbf{M}_{\mathbf{x}}^{-1} \mathbf{x}$  can be expressed as a quadratic form in terms of the parameter vector  $\mathbf{a}^e = [1 \ -\mathbf{a}]$ :

$$\mathbf{x}^T \mathbf{M}_{\mathbf{x}}^{-1} \mathbf{x} = \mathbf{a}^{eT} \mathbf{D} \mathbf{a}^e \quad (\text{C.9})$$

and Box, Jenkins and Reinsel [21] show that the elements of  $\mathbf{D}$  are symmetric sums of squared and lagged products of the data elements. This would appear to be helpful for obtaining exact ML estimates for the AR parameters since this exponent can easily be minimised w.r.t.  $\mathbf{a}$ . Note however that the determinant term  $|\mathbf{M}_{\mathbf{x}_0}|$  of the exact likelihood expression is variable with  $\mathbf{a}$  and its differentials w.r.t.  $\mathbf{a}$  are complicated functions of the AR coefficients, making the exact ML solution intractable to analytic solution.

# Appendix D

## Derivation of Likelihood for $\mathbf{i}$

In this appendix the likelihood  $p(\mathbf{y} | \mathbf{i})$  for a particular noise configuration  $\mathbf{i}$  is derived for general noise amplitude and data models. Consider firstly the PDF for  $\mathbf{n}$ , the additive noise component for a particular data block of length  $N$ . Wherever  $i_m = 0$  we know with certainty 1 that the noise component is zero. Otherwise  $i_m = 1$  and the noise takes some random amplitude defined by the noise burst amplitude PDF  $p_{\mathbf{n}(\mathbf{i})|\mathbf{i}}$  which is the joint distribution for noise amplitudes at the  $l$  sample locations where  $i_m = 1$ . Overall we have:

$$p_{\mathbf{n}}(\mathbf{n}) = \delta_{(N-l)}(\mathbf{n}_{-(\mathbf{i})}) p_{\mathbf{n}(\mathbf{i})|\mathbf{i}}(\mathbf{n}_{(\mathbf{i})} | \mathbf{i}) \quad (\text{D.1})$$

where  $\delta_{(k)}()$  is the  $k$ -dimensional delta function with unity volume and  $_{(\mathbf{i})}$  and  $_{-(\mathbf{i})}$  are the partitions as defined before. This delta function expresses the certainty that noise amplitudes are zero wherever  $i_m = 0$ .

We have the additive relationship

$$\mathbf{y} = \mathbf{x} + \mathbf{n}$$

for the data block. If the noise  $\mathbf{n}$  is assumed statistically independent of the data  $\mathbf{x}$  the PDF for  $\mathbf{y}$  conditional upon  $\mathbf{x}$  may be directly expressed as:

$$p(\mathbf{y} | \mathbf{x}, \mathbf{i}) = p_{\mathbf{n}}(\mathbf{y} - \mathbf{x}) = \delta_{(N-l)}(\mathbf{y}_{-(\mathbf{i})} - \mathbf{x}_{-(\mathbf{i})}) p_{\mathbf{n}(\mathbf{i})|\mathbf{i}}(\mathbf{y}_{(\mathbf{i})} - \mathbf{x}_{(\mathbf{i})}) \quad (\text{D.2})$$

For a particular data model with PDF  $p_{\mathbf{x}}$  the joint probability for data and observations is then given by

$$p(\mathbf{y}, \mathbf{x} | \mathbf{i}) = p(\mathbf{y} | \mathbf{x}, \mathbf{i}) p_{\mathbf{x}}(\mathbf{x}) \quad (\text{D.3})$$

$$= \delta_{(N-l)}(\mathbf{y}_{-(\mathbf{i})} - \mathbf{x}_{-(\mathbf{i})}) p_{\mathbf{n}(\mathbf{i})|\mathbf{i}}(\mathbf{y}_{(\mathbf{i})} - \mathbf{x}_{(\mathbf{i})} | \mathbf{i}) p_{\mathbf{x}}(\mathbf{x}) \quad (\text{D.4})$$

where we have used the assumption that the noise generating process is independent of the data to assign  $p_{\mathbf{x}|\mathbf{i}} = p_{\mathbf{x}}$ .

The likelihood is now obtained by a marginalisation integral over  $\mathbf{x}$ :

$$p(\mathbf{y} | \mathbf{i}) = \int_{\mathbf{x}} \delta_{(N-l)}(\mathbf{y}_{-(i)} - \mathbf{x}_{-(i)}) p_{\mathbf{n}(i)|\mathbf{i}}(\mathbf{y}_{(i)} - \mathbf{x}_{(i)} | \mathbf{i}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (\text{D.5})$$

and the integration may be immediately simplified using the sifting property of the delta function as:

$$p(\mathbf{y} | \mathbf{i}) = \int_{\mathbf{x}_{(i)}} p_{\mathbf{n}(i)|\mathbf{i}}(\mathbf{y}_{(i)} - \mathbf{x}_{(i)} | \mathbf{i}) p_{\mathbf{x}}(\mathbf{x}) |_{\mathbf{x}_{-(i)}=\mathbf{y}_{-(i)}} d\mathbf{x}_{(i)} \quad (\text{D.6})$$

$$= \int_{\mathbf{x}_{(i)}} p_{\mathbf{n}(i)|\mathbf{i}}(\mathbf{y}_{(i)} - \mathbf{x}_{(i)} | \mathbf{i}) p_{\mathbf{x}}(\mathbf{x}_{(i)}, \mathbf{y}_{-(i)}) d\mathbf{x}_{(i)} \quad (\text{D.7})$$

where  $p_{\mathbf{x}}(\mathbf{x}_{(i)}, \mathbf{y}_{-(i)}) = p_{\mathbf{x}}(\mathbf{x}) |_{\mathbf{x}_{-(i)}=\mathbf{y}_{-(i)}}$  is simply  $p_{\mathbf{x}}$  with observed data  $\mathbf{y}_{-(i)}$  substituted at the appropriate locations.

This expression for evidence can now be evaluated by substitution of the relevant functional forms for  $p_{\mathbf{x}}$  and  $p_{\mathbf{n}|\mathbf{i}}$ , both of which will be determined from modelling considerations.

## D.1 Gaussian noise bursts

Substituting the Gaussian noise PDF(9.10) and Gaussian AR data PDF (9.8) into the likelihood expression of equation (9.5) we arrive at the following integral:

$$p(\mathbf{y} | \mathbf{i}) = k \int_{\mathbf{x}_{(i)}} \exp\left(-\frac{1}{2} \left(a + \mathbf{b}^T \mathbf{x}_{(i)} + \mathbf{x}_{(i)}^T \mathbf{C} \mathbf{x}_{(i)}\right)\right) d\mathbf{x}_{(i)} \quad (\text{D.8})$$

where the terms  $k, a, \mathbf{b}, \mathbf{C}$  are defined as

$$k = \frac{1}{(2\pi)^{l/2} |\mathbf{R}_{\mathbf{n}(i)}|^{1/2} (2\pi\sigma_e^2)^{(N-P)/2}} \quad (\text{D.9})$$

$$a = \frac{\mathbf{y}_{-(i)}^T \mathbf{A}_{-(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)}}{\sigma_e^2} + \mathbf{y}_{(i)}^T \mathbf{R}_{\mathbf{n}(i)}^{-1} \mathbf{y}_{(i)} \quad (\text{D.10})$$

$$\mathbf{b} = \frac{2\mathbf{A}_{(i)}^T \mathbf{A}_{-(i)} \mathbf{y}_{-(i)}}{\sigma_e^2} - 2 \mathbf{R}_{\mathbf{n}(i)}^{-1} \mathbf{y}_{(i)} \quad (\text{D.11})$$

$$\mathbf{C} = \frac{\mathbf{A}_{(i)}^T \mathbf{A}_{(i)}}{\sigma_e^2} + \mathbf{R}_{\mathbf{n}(i)}^{-1}. \quad (\text{D.12})$$

Note that in equation (D.8)  $p(\mathbf{x}_0)$  has been placed outside the integral and treated as a multiplicative constant since, as discussed in the main text, the first  $P$  samples of the block are assumed to be uncorrupted. The integrand is of the same form as (A.3) and hence result (A.5) applies.

Substituting for  $k, a, \mathbf{b}$  and  $\mathbf{C}$  gives equations (9.11)-(9.16)

# Appendix E

## Marginalised Bayesian Detector

The likelihood calculations of (9.19)-(9.24) are implicitly conditional upon the AR coefficients  $\mathbf{a}$ , the excitation variance  $\sigma_e^2$ , the noise variance  $\sigma_n^2$ , the first  $P$  data samples  $\mathbf{x}_0$  (through use of the conditional likelihood) and all remaining modelling assumptions  $\mathcal{M}$ . For fixed  $\sigma_e^2$ ,  $\mathbf{a}$  and  $\mathbf{x}_0$  it was noted that the PDF  $p(\mathbf{x}_0)$  (C.3) was a constant which could be neglected. If  $\sigma_e^2$  is allowed to vary, this term is no longer a constant and must be accounted for. The exact likelihood expression (4.55) can be used to overcome this difficulty. We use the conditional likelihood expression for compatibility with the results of chapter 9.

The likelihood expression of (9.19) is then rewritten as

$$p(\mathbf{y} \mid \mathbf{i}, \sigma_e^2, \mu, \mathbf{a}, \mathcal{M}) \propto \frac{\mu^l \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}{(2\pi\sigma_e^2)^{\frac{N}{2}} |\Phi|^{1/2}} \quad (\text{E.1})$$

while results (9.20)-(9.24) remain unchanged.

We might wish to marginalise  $\sigma_e^2$  from the evidence. Note, however, that  $\mu = \frac{\sigma_e}{\sigma_n}$ , which means that (9.20)-(9.24) are strongly dependent on  $\sigma_e$ . If we are prepared to assume that  $\mu$  is independent of  $\sigma_e$  then these dependencies disappear. This assumption implies that the click variance  $\sigma_n^2$  is determined by a constant  $\frac{1}{\mu^2}$  times the excitation variance, i.e. click amplitudes are scaled *relative* to excitation energy. There is no reason to assume that noise variances should be related to the signal excitation in the case of degraded audio sources. It is also debatable as to whether marginalisation is preferable to long-term adaptive estimation of the unknown parameters

beforehand. However, it is hoped that marginalisation under this assumption may give improved robustness when the value of  $\sigma_e$  is uncertain.

Supposing  $\sigma_e^2$  is reparameterised as  $\lambda = \frac{1}{\sigma_e^2}$  and  $\lambda$  is independent of  $\mathbf{i}$ ,  $\mu$ ,  $\mathbf{a}$  and the modelling assumptions, then  $p(\lambda | \mathbf{i}, \mu, \mathbf{a}, \mathcal{M}) = p(\lambda)$ , and the joint distribution is given by:

$$p(\mathbf{y}, \lambda | \mathbf{i}, \mu, \mathbf{a}, \mathcal{M}) \propto p(\lambda) \frac{\mu^l \exp\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}{(2\pi\sigma_e^2)^{\frac{N}{2}} |\mathbf{\Phi}|^{1/2}} \quad (\text{E.2})$$

Suppose  $\lambda \sim G(\alpha, \beta)$ , i.e.  $Y$  is drawn from the two-parameter Gamma density (see appendix A). Then the joint posterior density of (E.2) is:

$$p(\mathbf{y}, \lambda | \mathbf{i}, \mu, \mathbf{a}, \mathcal{M}) \propto \lambda^{(N/2+\alpha-1)} \frac{\mu^l \exp(-\lambda(\beta + E_{\text{MIN}}/2))}{|\mathbf{\Phi}|^{1/2}} \quad (\text{E.3})$$

Considered as a function of  $\lambda$  this expression is proportional to  $G(\alpha + N/2, \beta + E_{\text{MIN}}/2)$ . Using result A.7 for the integration of such functions we obtain the following marginalised evidence expression:

$$p(\mathbf{y} | \mathbf{i}, \mu, \mathbf{a}, \mathcal{M}) = \int_{\lambda} p(\mathbf{y}, \lambda | \mathbf{i}, \mu, \mathbf{a}, \mathcal{M}) d\lambda \propto \frac{\mu^l (\beta + E_{\text{MIN}}/2)^{-(\alpha+N/2)}}{|\mathbf{\Phi}|^{1/2}} \quad (\text{E.4})$$

This expression can be used in place of (9.19) for detection. Values of  $\alpha$  and  $\beta$  must be selected beforehand. If a long-term estimate  $\hat{\sigma}_e^2$  of  $\sigma_e^2$  is available it will be reasonable to choose  $\alpha$  and  $\beta$  using (A.8)-(A.10) such that the mean or mode of the Gamma density is  $\hat{\lambda} = \frac{1}{\hat{\sigma}_e^2}$  and the covariance of the density expresses how confident we are in the estimate  $\hat{\lambda}$ . Initial results indicate that such a procedure gives improved detection performance when values of  $\sigma_e^2$  and  $\mu$  are uncertain.

If a long-term estimate of  $\sigma_e^2$  is not available the general Gamma prior may be inappropriate. Comparison of (E.2) with (E.3) shows that a uniform or Jeffreys' prior can be implemented within this framework. In particular,  $\alpha = 1, \beta \rightarrow 0$  corresponds to a uniform prior on  $\lambda$ , while  $\alpha = 0, \beta \rightarrow 0$  gives a Jeffreys' prior  $p(\lambda) = 1/\lambda$ .

Under certain assumptions we have shown that it is possible to marginalise  $\sigma_e^2$  from the evidence expression. It is not, however, straightforward to eliminate  $\mu$ , since this appears in the matrix expressions (9.20)-(9.24). The only case where marginalisation of  $\mu$  is straightforward is when the approximation of (9.25)-(9.27) is made, i.e.  $\mu$  is very small. In this case a second marginalisation integral can be performed to eliminate  $\mu$  from (E.4). Evaluation of such a procedure is left as future work.



# Appendix F

## Derivation of Sequential Update Formulae

This appendix derives from first principles the sequential update for the detection likelihood in chapter 10

Consider firstly how the  $(n - P) \times n$  matrix  $\mathbf{A}_n$  (see 4.53) is modified with a new input sample  $y_{n+1}$ . When the data block increases in length by one sample, matrix  $\mathbf{A}_n$  (by inspection) will have one extra row and column appended to it as defined by:

$$\mathbf{A}_{n+1} = \begin{bmatrix} \mathbf{A}_n & \mathbf{0}_n \\ -\mathbf{b}_n^T & 1 \end{bmatrix} \quad (\text{F.1})$$

where  $\mathbf{b}_n$  is defined as

$$\mathbf{b}_n = \begin{bmatrix} \mathbf{0}_{(n-P)}^T & a_P & a_{P-1} & \dots & a_2 & a_1 \end{bmatrix}^T \quad (\text{F.2})$$

and  $\mathbf{0}_q$  is the column vector containing  $q$  zeros.

$\mathbf{b}_n$  is a length  $n$  column vector and can thus be partitioned just as  $\mathbf{y}_n$  and  $\mathbf{x}_n$  into sections  $\mathbf{b}_{(i)n}$  and  $\mathbf{b}_{-(i)n}$  corresponding to unknown and known data samples for detection state estimate  $\mathbf{i}_n$ .

### F.1 Update for $i_{n+1} = 0$ .

Consider firstly how  $\mathbf{A}_{(i)n}$  and  $\mathbf{A}_{-(i)n}$  are updated when  $i_{n+1} = 0$ . In this case the incoming sample  $y_{n+1}$  is treated as uncorrupted. Hence the true

data sample  $x_{n+1}$  is considered known and  $\mathbf{A}_{(i)(n+1)}$  is obtained by the addition of just a single row to  $\mathbf{A}_{(i)n}$ :

$$\mathbf{A}_{(i)(n+1)} = \begin{bmatrix} \mathbf{A}_{(i)n} \\ -\mathbf{b}_{(i)n}^T \end{bmatrix}, \quad (\text{F.3})$$

while  $\mathbf{A}_{-(i)(n+1)}$  is obtained by the addition of both a row and a column:

$$\mathbf{A}_{-(i)(n+1)} = \begin{bmatrix} \mathbf{A}_{-(i)n} & \mathbf{0}_n \\ -\mathbf{b}_{-(i)n}^T & 1 \end{bmatrix}. \quad (\text{F.4})$$

In addition the partitioned input data vector is straightforwardly updated as:

$$\mathbf{y}_{(i)(n+1)} = \mathbf{y}_{(i)n}, \quad \mathbf{y}_{-(i)(n+1)} = [\mathbf{y}_{-(i)n}^T \ y_{n+1}]^T \quad (\text{F.5})$$

We now proceed to update  $\Phi_n$  (10.4). The first term is updated by direct multiplication of (F.3):

$$\begin{aligned} \mathbf{A}_{(i)(n+1)}^T \mathbf{A}_{(i)(n+1)} &= \mathbf{A}_{(i)n}^T \mathbf{A}_{(i)n} \\ &\quad + \mathbf{b}_{(i)n} \mathbf{b}_{(i)n}^T \end{aligned} \quad (\text{F.6})$$

The second term of  $\Phi_n$  is unchanged by the input of  $y_{n+1}$ . Hence  $\Phi_n$  is updated as:

$$\Phi_{(n+1)} = \Phi_n + \mathbf{b}_{(i)n} \mathbf{b}_{(i)n}^T \quad (\text{F.7})$$

The first term of  $\theta_n$  (10.5) is updated similarly using direct multiplication of results (F.3), (F.4), (F.5) to give:

$$\begin{aligned} \mathbf{A}_{(i)(n+1)}^T \mathbf{A}_{-(i)(n+1)} \mathbf{y}_{-(i)(n+1)} &= \mathbf{A}_{(i)n}^T \mathbf{A}_{-(i)n} \mathbf{y}_{-(i)n} \\ &\quad + \mathbf{b}_{(i)n} \left( \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} - y_{n+1} \right) \end{aligned} \quad (\text{F.8})$$

As for  $\Phi_n$ , the second term of  $\theta_n$  is unchanged for  $i_{n+1} = 0$ . Hence the update for  $\theta_n$  is:

$$\theta_{(n+1)} = \theta_n + \mathbf{b}_{(i)n} \left( y_{n+1} - \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \right) \quad (\text{F.9})$$

$$= \theta_n + \mathbf{b}_{(i)n} d_{n+1} \quad (\text{F.10})$$

where

$$d_{n+1} = y_{n+1} - \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \quad (\text{F.11})$$

The recursive updates derived for  $\Phi_n$  (F.7) and  $\theta_n$  (F.10) are now in the form required for application of the results given in appendix B (equations

(B.10)-(B.15)). Use of these results leads to recursive updates for the terms in the state likelihood (9.37), i.e.  $\mathbf{x}_{(i)n}^{\text{MAP}}$ ,  $E_{\text{MIN}n}$  and  $|\Phi_n|$ .

Now, defining

$$\mathbf{P}_n = \Phi_n^{-1}, \quad (\text{F.12})$$

the update for  $i_{n+1} = 0$  is given by

$$\mathbf{k}_{n+1} = \frac{\mathbf{P}_n \mathbf{b}_{(i)n}}{1 + \mathbf{b}_{(i)n}^T \mathbf{P}_n \mathbf{b}_{(i)n}} \quad (\text{F.13})$$

$$d_{n+1} = y_{n+1} - \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \quad (\text{F.14})$$

$$\alpha_{n+1} = d_{n+1} - \mathbf{x}_{(i)n}^{\text{MAP}T} \mathbf{b}_{(i)n} \quad (\text{F.15})$$

$$\mathbf{x}_{(i)(n+1)}^{\text{MAP}} = \mathbf{x}_{(i)n}^{\text{MAP}} + \mathbf{k}_{n+1} \alpha_{n+1} \quad (\text{F.16})$$

$$\mathbf{P}_{n+1} = \mathbf{P}_n - \mathbf{k}_{n+1} \mathbf{b}_{(i)n}^T \mathbf{P}_n \quad (\text{F.17})$$

$$|\mathbf{P}_{n+1}| = |\mathbf{P}_n| \left(1 + \mathbf{b}_{(i)n}^T \mathbf{P}_n \mathbf{b}_{(i)n}\right)^{-1} \quad (\text{F.18})$$

$$E_{\text{MIN}(n+1)} = E_{\text{MIN}n} + \alpha_{n+1} \left(d_{n+1} - \mathbf{x}_{(i)(n+1)}^{\text{MAP}T} \mathbf{b}_{(i)n}\right) \quad (\text{F.19})$$

$$\begin{aligned} p(\mathbf{y}_{(n+1)} | \mathbf{i}_{(n+1)}) &= p(\mathbf{y}_n | \mathbf{i}_n) \frac{1}{\sqrt{2\pi\sigma_e^2}} \left(1 - \mathbf{b}_{(i)n}^T \mathbf{k}_{n+1}\right)^{1/2} \\ &\quad \exp\left(-\frac{1}{2\sigma_e^2} \alpha_{n+1} \left(d_{n+1} - \mathbf{x}_{(i)(n+1)}^{\text{MAP}T} \mathbf{b}_{(i)(n)}\right)\right) \end{aligned} \quad (\text{F.20})$$

Note that the values of  $|\mathbf{P}_{n+1}|$ ,  $\mathbf{y}_{-(i)n}$  and  $E_{\text{MIN}(n+1)}$  are not specifically required for the likelihood update given in (F.20), but are included for completeness and clarity. In fact the update only requires storage of the terms  $\mathbf{P}_n$ ,  $\mathbf{x}_{(i)n}^{\text{MAP}}$  and the previous likelihood value  $p(\mathbf{y}_n | \mathbf{i}_n)$  in order to perform the full evidence update when  $y_{n+1}$  is input.

## F.2 Update for $i_{n+1} = 1$ .

As stated above, when  $i_n = 1$  the length of the MAP data estimate  $\mathbf{x}_{(i)(n+1)}^{\text{MAP}}$  is one higher than at the previous sample number. Correspondingly the order of  $\Phi_{(n+1)}$  and  $\theta_{(n+1)}$  are also one higher. This step-up in solution order is achieved in two stages. The first stage is effectively a prediction step which predicts the MAP data vector without knowledge of  $y_{n+1}$ . The second stage corrects this prediction based on the input sample  $y_{n+1}$ .

We follow a similar derivation to the  $i_{n+1} = 0$  case. Matrices  $\mathbf{A}_{(i)n}$  and  $\mathbf{A}_{-(i)n}$  are updated as follows (c.f. (F.3)) and (F.4)):

$$\mathbf{A}_{(i)(n+1)} = \begin{bmatrix} \mathbf{A}_{(i)n} & \mathbf{0}_n \\ -\mathbf{b}_{(i)n}^T & 1 \end{bmatrix} \quad (\text{F.21})$$

and

$$\mathbf{A}_{-(i)(n+1)} = \begin{bmatrix} \mathbf{A}_{-(i)n} \\ -\mathbf{b}_{-(i)n}^T \end{bmatrix} \quad (\text{F.22})$$

The input data partitioning is now given by (c.f. (F.5)):

$$\mathbf{y}_{(i)(n+1)} = [\mathbf{y}_{(i)n}^T y_{n+1}]^T \quad \mathbf{y}_{-(i)(n+1)} = \mathbf{y}_{-(i)n} \quad (\text{F.23})$$

A similar procedure of direct multiplication of the required terms leads to the following updates:

$$\mathbf{A}_{(i)(n+1)}^T \mathbf{A}_{(i)(n+1)} = \begin{bmatrix} (\mathbf{A}_{(i)n}^T \mathbf{A}_{(i)n} + \mathbf{b}_{(i)n} \mathbf{b}_{(i)n}^T) & -\mathbf{b}_{(i)n} \\ -\mathbf{b}_{(i)n}^T & 1 \end{bmatrix} \quad (\text{F.24})$$

and

$$\begin{aligned} & \mathbf{A}_{(i)(n+1)}^T \mathbf{A}_{-(i)(n+1)} \mathbf{y}_{-(i)(n+1)} \\ &= \begin{bmatrix} \mathbf{A}_{(i)n}^T \mathbf{A}_{-(i)n} \mathbf{y}_{-(i)n} + \mathbf{b}_{(i)n} \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \\ \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \end{bmatrix} \end{aligned} \quad (\text{F.25})$$

By inspection the updates to  $\Phi_n$  (10.4) and  $\theta_n$  (10.5) are now given by:

$$\Phi_{(n+1)} = \begin{bmatrix} (\Phi_n + \mathbf{b}_{(i)n} \mathbf{b}_{(i)n}^T) & -\mathbf{b}_{(i)n} \\ -\mathbf{b}_{(i)n}^T & 1 + \mu^2 \end{bmatrix} \quad (\text{F.26})$$

and

$$\theta_{(n+1)} = \begin{bmatrix} \theta_n - \mathbf{b}_{(i)n} \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \\ \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} + \mu^2 y_{n+1} \end{bmatrix} \quad (\text{F.27})$$

Various approaches are possible for updating for  $\mathbf{x}_{(i)n}^{\text{MAP}}$ , etc. The approach suggested here is in two stages. It is efficient to implement and has a useful physical interpretation. The first step is effectively a prediction step, in which the unknown data sample  $x_{n+1}$  is predicted from the past data based on the model and without knowledge of the corrupted input sample  $y_{n+1}$ . The second stage corrects the prediction to incorporate  $y_{n+1}$ .

In the prediction step we solve for  $\mathbf{x}_{(i)(n+1)}^{\text{PRED}}$  using:

$$\begin{aligned} & \begin{bmatrix} (\Phi_n + \mathbf{b}_{(i)n} \mathbf{b}_{(i)n}^T) & -\mathbf{b}_{(i)n} \\ -\mathbf{b}_{(i)n}^T & 1 \end{bmatrix} \mathbf{x}_{(i)(n+1)}^{\text{PRED}} \\ &= \begin{bmatrix} \theta_n - \mathbf{b}_{(i)n} \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \\ \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \end{bmatrix} \end{aligned} \quad (\text{F.28})$$

or equivalently

$$\Phi_{n+1}^{\text{PRED}} \mathbf{x}_{(i)(n+1)}^{\text{PRED}} = \theta_{n+1}^{\text{PRED}} \quad (\text{F.29})$$

It is easily verified by direct multiplication and use of the identity  $\Phi_n \mathbf{x}_{(i)n}^{\text{MAP}} = \theta_n$  that the prediction estimate  $\mathbf{x}_{(i)(n+1)}^{\text{PRED}}$  is given by:

$$\mathbf{x}_{(i)(n+1)}^{\text{PRED}} = \left[ \begin{array}{c} \mathbf{x}_{(i)n}^{\text{MAP}} \\ \mathbf{b}_{(i)n}^T \mathbf{x}_{(i)n}^{\text{MAP}} + \mathbf{b}_{-(i)n}^T \mathbf{y}_{-(i)n} \end{array} \right] \quad (\text{F.30})$$

This is exactly as should be expected, since the last element of  $\mathbf{x}_{(i)(n+1)}^{\text{PRED}}$  is simply the single step linear prediction with values from  $\mathbf{x}_{(i)n}^{\text{MAP}}$  substituted for unknown samples. Since the linear prediction will always have zero excitation the previous missing samples remain unchanged at their last value  $\mathbf{x}_{(i)n}^{\text{MAP}}$ .

The inverse of matrix  $\Phi_{n+1}^{\text{PRED}}$  is easily obtained from the result for the inverse of a partitioned matrix (see B.3) as:

$$\Phi_{(n+1)}^{\text{PRED}^{-1}} = \left[ \begin{array}{cc} \Phi_n^{-1} & \Phi_n^{-1} \mathbf{b}_{(i)n} \\ \mathbf{b}_{(i)n}^T \Phi_n^{-1} & \left( 1 + \mathbf{b}_{(i)n}^T \Phi_n^{-1} \mathbf{b}_{(i)n} \right) \end{array} \right] \quad (\text{F.31})$$

all of whose terms are known already from the update for  $i_{n+1} = 0$ . The determinant of  $\Phi_{n+1}^{\text{PRED}}$  can be shown using result (B.5) to be unchanged by the update. The error energy term  $E_{\text{MIN}(n+1)}$  is also unaffected by this part of the update, since the single sample data prediction adds nothing to the excitation energy.

The second stage of the update involves the correction to account for the new input data sample  $y_{n+1}$ . By comparison of (F.26) and (F.27) with (F.28) and F.29) we see that the additional update to give  $\Phi_{n+1}$  and  $\theta_{n+1}$  is:

$$\Phi_{n+1} = \Phi_{n+1}^{\text{PRED}} + \left[ \begin{array}{cc} \mathbf{0}_{n \times n} & \mathbf{0}_n \\ \mathbf{0}_n^T & \mu^2 \end{array} \right] \quad (\text{F.32})$$

and

$$\theta_{n+1} = \theta_{n+1}^{\text{PRED}} + \left[ \begin{array}{c} \mathbf{0}_n \\ \mu^2 y_{n+1} \end{array} \right] \quad (\text{F.33})$$

where  $\mathbf{0}_{n \times n}$  is the all-zero matrix of dimension  $(n \times n)$ . This form of update can be performed very efficiently using a special case of the extended RLS equations given in (B.10) (B.15) in which ‘input’ vector  $\mathbf{u}_n$  and ‘desired signal’  $d_{n+1}$  are given by:

$$\mathbf{u}_n = [\mathbf{0}_n^T \mu]^T \quad \text{and} \quad d_{n+1} = \mu y_{n+1} \quad (\text{F.34})$$

$\Phi_{n+1}^{\text{PRED}}$  is associated with  $\Phi_n$  in the appendix and  $\mathbf{x}_{n+1}^{\text{PRED}}$  with  $\mathbf{x}_n$ . Significant simplifications follow directly as a result of  $\mathbf{u}_n$  having only one non-zero

element. The majority of the terms required to make this second stage of update will already have been calculated during the update for  $i_{n+1} = 0$  and they will thus not generally need to be re-calculated. Section 10.2.1 in the main text summarises the full update for both  $i_{n+1} = 0$  and  $i_{n+1} = 1$  and explicitly shows which terms are common to both cases.

# Appendix G

## Derivations for EM-based Interpolation

This appendix derives the EM update equations for the Bayesian interpolation problem of chapter 12. Section 4.5 summarises the general form of the EM algorithm. The first stage calculates an expectation of the log augmented posterior  $p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{y})$  conditional upon the current estimate  $\mathbf{z}^i$ . A notation to make the expectations clear is introduced such that:

$$E_\phi[x|y] = \int_\phi x p(\phi|y) d\phi.$$

We define the latent data  $\mathbf{z} = \mathbf{x}_{(i)}$  for this derivation. For the interpolation model the expectation step can be split into separate expectations over noise and signal parameters  $\boldsymbol{\theta}_v$  and  $\boldsymbol{\theta}_x$ :

$$\begin{aligned} Q(\mathbf{z}, \mathbf{z}^i) &= E_\theta[\log(p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{y})|\mathbf{z}^i, \mathbf{y})] & (G.1) \\ &= E_\theta[\log(p(\mathbf{z}, \mathbf{y}|\boldsymbol{\theta})) - \log(p(\mathbf{y}|\boldsymbol{\theta}))|\mathbf{z}^i, \mathbf{y}] \\ &= E_\theta[\log(p(\mathbf{z}, \mathbf{y}|\boldsymbol{\theta}))|\mathbf{z}^i, \mathbf{y}] - C \\ &= E_\theta[\log(p(\mathbf{x}|\boldsymbol{\theta}_x)) + \log(p(\mathbf{v}_{(i)}|\boldsymbol{\theta}_v))|\mathbf{z}^i, \mathbf{y}] - C \\ &= E_{\boldsymbol{\theta}_x}[\log(p(\mathbf{x}|\boldsymbol{\theta}_x))|\mathbf{x}^i] + E_{\boldsymbol{\theta}_v}[\log(p(\mathbf{v}_{(i)}|\boldsymbol{\theta}_v))|\mathbf{v}^i] - C \\ &= E_x + E_v - C & (G.2) \end{aligned}$$

where  $C$  is constant since  $p(\mathbf{y}|\boldsymbol{\theta})$  does not depend on  $\mathbf{z}$ . In the third line  $p(\mathbf{z}, \mathbf{y}|\boldsymbol{\theta})$  has been expanded as:

$$\begin{aligned} p(\mathbf{z}, \mathbf{y}|\boldsymbol{\theta}) &= p(\mathbf{x}_{(i)}, \mathbf{x}_{-(i)}, \mathbf{y}_{(i)}|\boldsymbol{\theta}) \\ &= p(\mathbf{x}|\boldsymbol{\theta}_x)p(\mathbf{v}_{(i)}|\boldsymbol{\theta}_v) \end{aligned}$$

We now present the signal and noise expectations,  $E_x$  and  $E_v$ .

### G.1 Signal expectation, $E_x$

The signal expectation is taken over the signal parameters  $\boldsymbol{\theta}_x = \{\mathbf{a}, \lambda_e\}$  conditional upon the current estimate of the reconstructed data  $\mathbf{x}^i$ .

The expectation over signal parameters can be rewritten as:

$$\begin{aligned} E_x &= E_{\boldsymbol{\theta}_x} [\log(p(\mathbf{x}|\boldsymbol{\theta}_x))|\mathbf{x}^i] \\ &= E_{\lambda_e} [E_{\mathbf{a}} [\log(p(\mathbf{x}|\boldsymbol{\theta}_x))|\lambda_e, \mathbf{x}^i] |\mathbf{x}^i] \\ &= E_{\lambda_e} [\lambda_e/2 E_{\mathbf{a}} [-E(\mathbf{x}, \mathbf{a})|\lambda_e, \mathbf{x}^i] |\mathbf{x}^i] + D \end{aligned} \quad (\text{G.3})$$

where  $D$  is a constant which does not depend upon  $\mathbf{x}$  and (12.6) has been substituted for  $p(\mathbf{x}|\boldsymbol{\theta}_x)$ . The resulting ‘nested’ expectations are taken w.r.t.  $p(\mathbf{a}|\lambda_e, \mathbf{x}^i)$  and  $p(\lambda_e|\mathbf{x}^i)$  which can be obtained using Bayes’ Theorem as:

$$\begin{aligned} p(\boldsymbol{\theta}_x|\mathbf{x}) &\propto p(\mathbf{x}|\boldsymbol{\theta}_x) p(\boldsymbol{\theta}_x) \\ &\propto \text{AR}_N(\mathbf{x}|\mathbf{a}, \lambda_e) G(\lambda_e|\alpha_e, \beta_e) \end{aligned} \quad (\text{G.4})$$

$$\begin{aligned} &= N_P(\mathbf{a}|\mathbf{a}^{\text{MAP}}(\mathbf{x}), (\lambda_e \mathbf{X}^T \mathbf{X}))^{-1} \\ &\quad G(\lambda_e|\alpha_e + (N - P)/2, \beta_e + E(\mathbf{x}, \mathbf{a}^{\text{MAP}}(\mathbf{x}))/2) \end{aligned} \quad (\text{G.5})$$

$$= p(\mathbf{a}|\lambda_e, \mathbf{x}) p(\lambda_e|\mathbf{x}) \quad (\text{G.6})$$

where the rearrangement between (G.4) and (G.5) is achieved by expanding and noting that the resulting density must be normalised w.r.t.  $\boldsymbol{\theta}_x$ , and the term  $\text{AR}_N(\mathbf{x}|\mathbf{a}, \lambda_e)$  denotes the conditional likelihood of the AR process, as defined in (12.6).  $\mathbf{a}^{\text{MAP}}(\mathbf{x})$  is defined as the MAP parameter estimate for data  $\mathbf{x}$ :  $\mathbf{a}^{\text{MAP}}(\mathbf{x}) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{x}_1$ .

The required densities  $p(\mathbf{a}|\lambda_e, \mathbf{x})$  and  $p(\lambda_e|\mathbf{x})$  can be directly identified from comparison of (G.5) and (G.6) as

$$p(\mathbf{a}|\lambda_e, \mathbf{x}) = N_P(\mathbf{a}|\mathbf{a}^{\text{MAP}}(\mathbf{x}), (\lambda_e \mathbf{X}^T \mathbf{X})^{-1})$$

$$\text{and} \quad p(\lambda_e|\mathbf{x}) = G(\alpha_e + (N - P)/2, \beta_e + E(\mathbf{x}, \mathbf{a}^{\text{MAP}}(\mathbf{x}))/2)$$

The inner expectation of (G.3) is now taken w.r.t.  $p(\mathbf{a}|\lambda_e, \mathbf{x}^i)$ , giving:

$$\begin{aligned} &E_{\mathbf{a}} [-E(\mathbf{x}, \mathbf{a})|\lambda_e, \mathbf{x}^i] \\ &= -\mathbf{x}_1^T \mathbf{x}_1 + 2\mathbf{x}_1^T \mathbf{X} E_{\mathbf{a}} [\mathbf{a}|\lambda_e, \mathbf{x}^i] - \text{trace}(\mathbf{X} E_{\mathbf{a}} [\mathbf{a}\mathbf{a}^T|\lambda_e, \mathbf{x}^i] \mathbf{X}^T) \\ &= -\mathbf{x}_1^T \mathbf{x}_1 + 2\mathbf{x}_1^T \mathbf{X} \mathbf{a}^i - \text{trace}\left(\mathbf{X} \left( (\lambda_e \mathbf{X}^{iT} \mathbf{X}^i)^{-1} + \mathbf{a}^i \mathbf{a}^{iT} \right) \mathbf{X}^T\right) \\ &= -E(\mathbf{x}, \mathbf{a}^i) - \frac{1}{\lambda_e} \text{trace}\left(\mathbf{X} (\mathbf{X}^{iT} \mathbf{X}^i)^{-1} \mathbf{X}^T\right) \\ &= -E(\mathbf{x}, \mathbf{a}^i) - T(\mathbf{x}, \mathbf{x}^i)/\lambda_e \end{aligned} \quad (\text{G.7})$$



In this expression the mean and covariance of the conditional density  $p(\mathbf{a}|\lambda_e, \mathbf{x}^i) = \mathbf{N}_P(\mathbf{a}^i, (\lambda_e \mathbf{X}^i{}^T \mathbf{X}^i)^{-1})$ , where  $\mathbf{a}^i = \mathbf{a}^{\text{MAP}}(\mathbf{x}^i)$ , lead directly to the result  $\mathbf{E}_{\mathbf{a}}[\mathbf{a}\mathbf{a}^T|\lambda_e, \mathbf{x}^i] = \mathbf{a}^i \mathbf{a}^i{}^T + (\lambda_e \mathbf{X}^i{}^T \mathbf{X}^i)^{-1}$ . Note that (G.7) is equivalent to the E-step derived in Ó Ruanaidh and Fitzgerald [167, 166] for missing data interpolation.

$T(\mathbf{x}, \mathbf{x}^i)$  can be written as a quadratic form in  $\mathbf{x}$ :

$$\begin{aligned} T(\mathbf{x}, \mathbf{x}^i) &= \text{trace} \left( \mathbf{X} \left( \mathbf{X}^i{}^T \mathbf{X}^i \right)^{-1} \mathbf{X}^T \right) \\ &= \mathbf{x}^T \mathbf{T}(\mathbf{x}^i) \mathbf{x} \\ \text{where } \mathbf{T}(\mathbf{x}^i) &= \sum_{q=1}^{N-P+1} \mathbf{N}_q(\mathbf{x}^i) \end{aligned} \quad (\text{G.8})$$

$\mathbf{N}_q(\mathbf{x}^i)$  is the all-zero  $(N \times N)$  matrix with the  $(P \times P)$  sub-matrix  $(\mathbf{X}^i{}^T \mathbf{X}^i)^{-1}$  substituted starting at the  $q$ th element of the leading diagonal:

$$\mathbf{N}_q(\mathbf{x}^i) = \begin{bmatrix} \mathbf{0}_{q-1, q-1} & \dots & \dots & \ddots \\ \vdots & (\mathbf{X}^i{}^T \mathbf{X}^i)^{-1} & \vdots & \vdots \\ \ddots & \dots & \mathbf{0}_{N-q-p+1, N-q-p+1} & \dots \end{bmatrix} \quad (\text{G.9})$$

where  $\mathbf{0}_{nm}$  denotes the  $(n \times m)$  all-zero matrix.  $\mathbf{T}(\mathbf{x}^i)$  is thus a band-diagonal matrix calculated as the summation of all  $N - P + 1$  configurations of  $\mathbf{N}_q(\mathbf{x}^i)$ . Note that the structure of  $\mathbf{T}(\mathbf{x}^i)$  is symmetrical throughout and almost Toeplitz except for its first and last  $P$  columns and rows.

The outer expectation of (G.3) is then taken w.r.t.  $p(\lambda_e|\mathbf{x}^i)$  to give:

$$\begin{aligned} E_x &= \mathbf{E}_{\lambda_e} [-\lambda_e E(\mathbf{x}, \mathbf{a}^i)/2 - T(\mathbf{x}, \mathbf{x}^i)/2|\mathbf{x}^i] + D \\ &= -\frac{E(\mathbf{x}, \mathbf{a}^i)(\alpha_e + (N - P)/2)}{2(\beta_e + E(\mathbf{x}^i, \mathbf{a}^i)/2)} - \frac{T(\mathbf{x}, \mathbf{x}^i)}{2} + D \\ &= -\frac{\lambda_e^i E(\mathbf{x}, \mathbf{a}^i)}{2} - \frac{T(\mathbf{x}, \mathbf{x}^i)}{2} + D \end{aligned} \quad (\text{G.10})$$

The term  $\lambda_e^i$  is the expected value of  $\lambda_e$  conditional upon  $\mathbf{x}^i$  and is obtained directly as the mean of the gamma distribution  $p(\lambda_e|\mathbf{x}^i)$ .

The term  $D$  does not depend upon  $\mathbf{x}$  and so may be ignored as an additive constant which will not affect the maximisation step.

## G.2 Noise expectation, $E_v$

The noise parameters are  $\boldsymbol{\theta}_v = \boldsymbol{\Lambda}_{(i)}$  where  $\boldsymbol{\Lambda} = [\lambda_1 \dots \lambda_N]^T$  and  $\lambda_t = 1/\sigma_{v_t}^2$ . In the common variance noise model we have that  $\sigma_{v_t}^2 = \sigma_v^2$ , a constant for all  $t$ . The noise parameter expectation depends now upon the form of noise model chosen:

1. **Independent variance.** The expectation is taken w.r.t.  $p(\boldsymbol{\theta}_v | \mathbf{v}_{(i)}^i)$ , which is given by:

$$\begin{aligned} p(\boldsymbol{\theta}_v | \mathbf{v}_{(i)}) &= p(\boldsymbol{\Lambda}_{(i)} | \mathbf{v}_{(i)}) \\ &\propto p(\mathbf{v}_{(i)} | \boldsymbol{\Lambda}_{(i)}) p(\boldsymbol{\Lambda}_{(i)}) \\ &\propto N_l(\mathbf{v}_{(i)} | \mathbf{0}, (\text{diag}(\boldsymbol{\Lambda}_{(i)}))^{-1}) \prod_{t \in \mathcal{I}} G(\lambda_{v_t} | \alpha_v, \beta_v) \\ &= \prod_{t \in \mathcal{I}} G(\lambda_{v_t} | \alpha_v + 1/2, \beta_v + v_t^2/2) \end{aligned} \quad (\text{G.11})$$

where we have substituted the noise likelihood and prior expressions and noted that the resultant densities must be normalised w.r.t.  $\lambda_{v_t}$ . Recall also that  $l$  is the number of degraded samples.

The noise expectation is then obtained as

$$\begin{aligned} E_v &= E_{\boldsymbol{\theta}_v} [\log(p(\mathbf{v}_{(i)} | \boldsymbol{\theta}_v)) | \mathbf{v}_{(i)}^i] \\ &= - \sum_{t \in \mathcal{I}} v_t^2 E_{\lambda_{v_t}} [\lambda_{v_t} | v_t^i] / 2 + G \\ &= - \sum_{t \in \mathcal{I}} v_t^2 \lambda_{v_t}^i / 2 + G \end{aligned} \quad (\text{G.12})$$

where  $\lambda_{v_t}^i = \frac{\alpha_v + 1/2}{\beta_v + v_t^2/2}$  is the expected value of the gamma distribution  $G(\lambda_{v_t} | \alpha_v + 1/2, \beta_v + v_t^2/2)$  (see appendix A.3).  $G$  is once again a constant which is independent of  $\mathbf{x}$ .

2. **Common variance.** A similar argument gives for the common variance model:

$$\begin{aligned} E_n &= E_{\lambda_v} \left[ - \sum_{t \in \mathcal{I}} \lambda_v v_t^2 / 2 | \mathbf{v}_{(i)}^i \right] + G \\ &= - \lambda_v^i \mathbf{v}_{(i)}^T \mathbf{v}_{(i)} / 2 + G \end{aligned} \quad (\text{G.13})$$

where  $\lambda_{v_t}^i = \lambda_v^i = (\alpha_n + l/2) / (\beta_n + \mathbf{v}_{(i)}^i{}^T \mathbf{v}_{(i)}^i / 2)$  is the expected value of the gamma distribution  $p(\lambda_v | \mathbf{v}_{(i)}^i)$ .

The two results can be summarised as:

$$E_n = - \frac{\mathbf{v}_{(i)}^T \mathbf{M}^i \mathbf{v}_{(i)}}{2} + G \quad (\text{G.14})$$

where  $\mathbf{M}^i = \text{diag}(\{\lambda_{v_t}^i; t \in \mathcal{I}\})$  is the expected value of the noise inverse-covariance matrix conditioned upon the current estimate of the noise,  $\mathbf{v}_{(i)}^i = \mathbf{y}_{(i)} - \mathbf{x}_{(i)}^i$ .  $G$  is an additive constant which does not depend upon  $\mathbf{x}$ .

### G.3 Maximisation step

The  $Q(.,.)$  function (G.2) is now obtained as the sum of  $E_x$  (G.10) and  $E_n$  (G.14). The individual components of  $Q(.,.)$  are all quadratic forms in the reconstructed data  $\mathbf{x}$  and hence also quadratic in  $\mathbf{z}$ , which is a sub-vector of  $\mathbf{x}$ . Maximisation of  $Q(.,.)$  is thus a linear operation, obtained as the solution of  $\frac{\partial Q(\mathbf{z}, \mathbf{z}^i)}{\partial \mathbf{z}} = \mathbf{0}$ :

$$\mathbf{z}^{i+1} = -\Psi^{-1} \left( \left( \lambda_e^i \mathbf{A}^{iT} \mathbf{A}^i + \mathbf{T}(\mathbf{x}^i) \right)_{(i)-(i)} \mathbf{y}_{-(i)} - \mathbf{M}^i \mathbf{y}_{(i)} \right) \quad (\text{G.15})$$

$$\text{where } \Psi = \left( \lambda_e^i \mathbf{A}^{iT} \mathbf{A}^i + \mathbf{T}(\mathbf{x}^i) \right)_{(i)(i)} + \mathbf{M}^i$$

and the notation ‘ $_{(i)(i)}$ ’ denotes the sub-matrix containing all elements whose row and column numbers are members of  $\mathcal{I}$ . Similarly, ‘ $_{(i)-(i)}$ ’ extracts a sub-matrix whose row numbers are in  $\mathcal{I}$  and whose column numbers are not in  $\mathcal{I}$ .

The complete iteration for EM interpolation is summarised in the main text.



# Appendix H

## Derivations for Gibbs Sampler

In this appendix results required for the Gibbs sampler detection and interpolation scheme are derived (see chapter 12).

### H.1 Gaussian/inverted-gamma scale mixtures

With the assumption of a zero mean normal distribution with unknown variance for a noise source  $v$ , the joint distribution for noise amplitude and noise variance is given by:

$$\begin{aligned}
 p(v, \sigma^2) &= p(v|\sigma^2)p(\sigma^2) \\
 &= \text{N}(v|0, \sigma^2)p(\sigma^2)
 \end{aligned}$$

The ‘effective’ distribution for  $v$  is then obtained by marginalisation of the joint distribution:

$$p(v) = \int_{\sigma^2} \text{N}(0, \sigma^2)p(\sigma^2)d\sigma^2$$

The convolving effect of this mixture process will generate non-Gaussian distributions whose properties depend on the choice of prior  $p(\sigma^2)$ . In the case of the IG prior, simple analytic results exist, based on the use of the

IG normalising constant in (A.4):

$$\begin{aligned}
p(v) &= \int_{\sigma^2} N(v|0, \sigma^2) \text{IG}(\sigma^2|\alpha, \beta) d\sigma^2 \\
&= \int_{\sigma^2} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-v^2/2\sigma^2) \\
&\quad \times \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-(\alpha+1)} \exp(-\beta/\sigma^2) d\sigma^2 \\
&= \frac{1}{\sqrt{2\pi}} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha + 1/2)}{(\beta + v^2/2)^{(\alpha+1/2)}} \\
&\quad \times \int_{\sigma^2} \text{IG}(\sigma^2|\alpha + 1/2, \beta + v^2/2) d\sigma^2 \\
&= \frac{1}{\sqrt{2\pi}} \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha + 1/2)}{(\beta + v^2/2)^{(\alpha+1/2)}} \\
&= \text{St}(v|0, \alpha/\beta, 2\alpha)
\end{aligned}$$

where  $\text{St}(\cdot)$  denotes the Student-t distribution:

$$\text{St}(x|\mu, \lambda, \alpha) = k(1 + (x - \mu)^2 \lambda / \alpha)^{-(\alpha+1)/2} \quad (\text{H.1})$$

The Student distribution, which has both Gaussian and Cauchy as limiting cases, is well known to be robust to impulses, since the ‘tails’ of the distribution decay algebraically, unlike the much less robust Gaussian with its exponentially decaying tails. The family of Student-t curves which arises as a result of the normal/inverted-gamma scale mixture is displayed in figure H.1, for the same set of  $\alpha$  and  $\beta$  parameters as in figure A.1. It can be seen that with suitable choice of  $\alpha$  and  $\beta$  the Student distribution can assign significant probability mass to high amplitude impulses which would have negligible mass in the corresponding normal distribution.

## H.2 Posterior distributions

### H.2.1 Joint posterior

The joint posterior distribution for all the unknowns is obtained from the likelihood, signal model and the prior distribution  $p(\boldsymbol{\theta})$ . Note that elements from the unknown parameter set  $\boldsymbol{\theta} = \{\mathbf{i}, \mathbf{a}, \sigma_e^2, \sigma_{v_t}^2 \ (t = 0 \dots N - 1)\}$  are assumed independent *a priori*, so that the prior can be expressed as:

$$p(\boldsymbol{\theta}) = p(\mathbf{i})p(\mathbf{a})p(\sigma_e^2) \prod_t p(\sigma_{v_t}^2) \quad (\text{H.2})$$

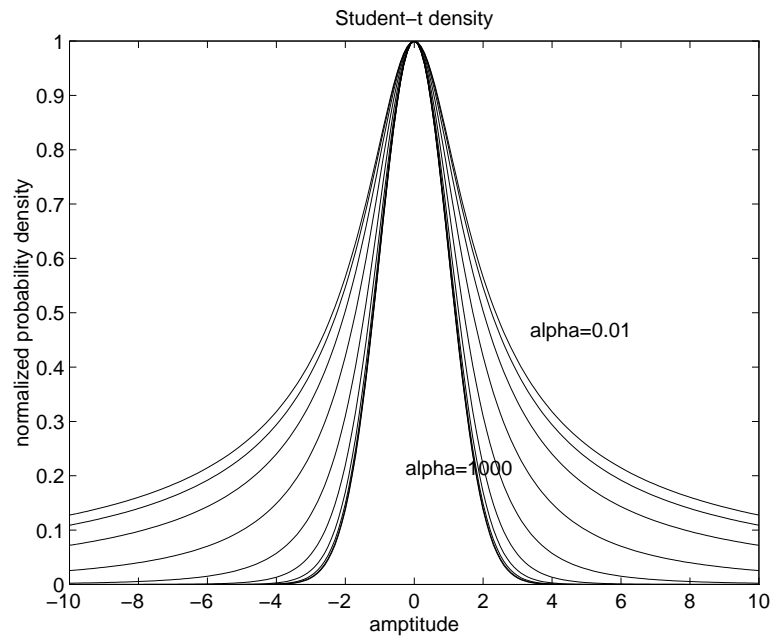


FIGURE H.1. Normal/Inverted-gamma mixture family (Student-t) with mode=1,  $\alpha = 0.01 \dots 1000$

which is simply the product of the individual prior terms for each parameter. The full posterior distribution is then obtained as:

$$\begin{aligned}
p(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) \propto & p(i_0) \prod_{\{t: i_t=0\}} \delta(y_t - x_t) \prod_{\{t: i_t=1\}} N(y_t - x_t | 0, \sigma_{v_t}^2) \\
& \times N_N(\mathbf{x} | \mathbf{0}, \sigma_e^2 (\mathbf{A}^T \mathbf{A})^{-1}) \\
& \times \prod_t P_{i_{t-1} i_t} \\
& \times \text{IG}(\sigma_e^2 | \alpha_e, \beta_e) \prod_t \text{IG}(\sigma_{v_t}^2 | \alpha_v, \beta_v)
\end{aligned} \tag{H.3}$$

### H.2.2 Conditional posteriors

The conditional posteriors are obtained by simple manipulation of the full posterior (H.3). For example, the conditional for  $\mathbf{a}$  is expressed as:

$$p(\mathbf{a} | \mathbf{x}, \boldsymbol{\theta}_{-(\mathbf{a})}, \mathbf{y}) = \frac{p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y})}{p(\mathbf{x}, \boldsymbol{\theta}_{-(\mathbf{a})} | \mathbf{y})}$$

where  $\boldsymbol{\theta}_{-(\mathbf{a})}$  denotes all elements of  $\boldsymbol{\theta}$  except for  $\mathbf{a}$ . Note, however, that the denominator term is simply a normalising constant for any given values of  $\mathbf{x}$ ,  $\boldsymbol{\theta}_{-(\mathbf{a})}$  and  $\mathbf{y}$ . The required conditional for  $\mathbf{a}$  can thus be obtained simply by grouping together all the terms in the joint posterior expression which depend upon  $\mathbf{a}$  and finding the appropriate normaliser to form a proper density. In the simplest cases this can be achieved by recognising that the terms depending on  $\mathbf{a}$  can be rewritten in the form of a well-known distribution. Here, the only term depending on  $\mathbf{a}$  is the signal model  $N_N(\mathbf{x} | \mathbf{0}, \sigma_e^2 (\mathbf{A}^T \mathbf{A})^{-1})$  which is easily rewritten using (12.4) to give the normalised conditional expression (12.11). Similar manipulations lead to the multivariate normal conditional for  $\mathbf{x}$  (12.14), obtained by rearranging the product of likelihood and signal model, both of which depend upon  $\mathbf{x}$ .

The noise variance terms can be treated similarly. For example,  $\sigma_e^2$  is present in both its own prior density  $\text{IG}(\sigma_e^2 | \alpha_e, \beta_e)$  and the signal model. Grouping together the exponential and power-law terms in  $\sigma_e^2$  leads to a conditional which is still in the general form of the  $\text{IG}(\cdot)$  density, as given in (12.12).

The detection indicator variables are discrete, but the same principles apply. In this case the conditional distribution has no well-known form and the normalising constant must be determined by evaluation of the joint posterior at all  $2^N$  possible values of  $\mathbf{i}$ . In this work, the indicator vector is treated jointly with the reconstructed data  $\mathbf{x}$ , as discussed more fully in section 12.5.2.



### H.3 Sampling the prior hyperparameters

**Noise variance parameters  $\alpha_v$  and  $\beta_v$ .** Conditional posterior distributions for the noise variance parameters  $\alpha_v$  and  $\beta_v$  can be obtained directly from the noise variance prior as follows:

$$\begin{aligned} p(\alpha_v, \beta_v | \mathbf{x}, \boldsymbol{\theta}_{-(\alpha_v, \beta_v)}, \mathbf{y}) &= p(\alpha_v, \beta_v | \sigma_{v_t}^2 \ (t = 0 \dots N-1)) \\ &\propto p(\sigma_{v_t}^2 \ (t = 0 \dots N-1) | \alpha_v, \beta_v) p(\alpha_v) p(\beta_v) \\ &\propto \prod_t \text{IG}(\sigma_{v_t}^2 | \alpha_v, \beta_v) p(\alpha_v) p(\beta_v) \end{aligned}$$

where we have assumed  $\alpha_v$  and  $\beta_v$  independent *a priori*

Rearrangement of this expression leads to the following univariate conditionals:

$$p(\alpha_v | \mathbf{x}, \boldsymbol{\theta}_{-(\alpha_v)}, \mathbf{y}) \propto \frac{\beta_v^{N\alpha_v} \Pi^{-(1+\alpha_v)}}{\Gamma(\alpha_v)^N} p(\alpha_v) \quad (\text{H.4})$$

and

$$p(\beta_v | \mathbf{x}, \boldsymbol{\theta}_{-(\beta_v)}, \mathbf{y}) = \text{G}(N\alpha_v + a, \Sigma + b) \quad (\text{H.5})$$

where  $\Sigma = \sum_t 1/\sigma_{v_t}^2$ ,  $\Pi = \prod_t \sigma_{v_t}^2$  and  $\text{G}(\cdot)$  denotes the Gamma distribution,  $p(x | \alpha, \beta) \propto x^{\alpha-1} \exp(-\beta x)$ , and we have assigned the prior  $p(\beta_v) = \text{G}(a, b)$ . Sampling from the distribution of equation (H.5) is a standard procedure. Non-informative or informative priors can be incorporated as appropriate by suitable choice of  $a$  and  $b$ . (H.4) is not a well known distribution and must be sampled by other means. This is not difficult, however, since the distribution is univariate. In any case, the precise value of  $\alpha_v$  is unlikely to be important, so we sample  $\alpha_v$  from a uniform grid of discrete values with probability mass given by (H.4). In principle any prior could easily be used for  $\alpha_v$  but we choose a uniform prior in the absence of any further information.

As an alternative to the above it may be reasonable to reparameterise  $\alpha_v$  and  $\beta_v$  in terms of variables with a more ‘physical’ interpretation, such as the mean or mode of the noise variance distribution, the idea being that it is usually easier to assign a prior distribution to such parameters. Since the mean of the  $\text{IG}(\cdot)$  distribution is only defined for  $\alpha_v > 1$  (see appendix A.4) we use the mode here, given by  $m_v = \beta_v/(\alpha_v + 1)$ . Another possibility would be the ratio  $\beta_v/\alpha_v$  which corresponds to the inverse precision parameter  $1/\lambda$  of the equivalent Student-t noise distribution (see appendix H.1).

Similar working to the above leads to the following conditionals for  $\alpha_v$  and  $m_v$ :

$$p(\alpha_v | \mathbf{x}, \boldsymbol{\theta}_{-(\alpha_v)}, \mathbf{y}) \propto \frac{(m_v(\alpha_v + 1))^{N\alpha_v} \Pi^{-(1+\alpha_v)} \exp(-(\alpha_v + 1)m_v \Sigma)}{\Gamma(\alpha_v)^N} p(\alpha_v) \quad (\text{H.6})$$

and

$$p(m_v|\mathbf{x}, \boldsymbol{\theta}_{-(m_v)}, \mathbf{y}) = \text{G}(N\alpha_v + a, (\alpha_v + 1)\Sigma + b) \quad (\text{H.7})$$

**Markov chain transition probabilities,  $P_{01}$  and  $P_{10}$ .** Conditionals for  $P_{01}$  and  $P_{10}$  are obtained in a similar fashion, since they are conditionally dependent only upon the switching values  $i_t$ :

$$\begin{aligned} p(P_{01}, P_{10}|\boldsymbol{\theta}_{-(P_{01}, P_{10})}, \mathbf{y}) &= p(P_{01}, P_{10}|\mathbf{i}) \\ &\propto p(\mathbf{i}|P_{01}, P_{10})p(P_{01})p(P_{10}) \\ &\propto p(P_{01})p(P_{10})p(i_0) \\ &\quad \prod_{t:i_t=1, i_{t-1}=0} P_{01} \prod_{t:i_t=0, i_{t-1}=0} (1 - P_{01}) \\ &\quad \prod_{t:i_t=0, i_{t-1}=1} P_{10} \prod_{t:i_t=1, i_{t-1}=1} (1 - P_{10}) \end{aligned}$$

Rearrangement of this expression under the assumption of uniform or beta distributed priors  $p(P_{01})$  and  $p(P_{10})$  leads to univariate conditionals which are in the form of the beta distribution,  $p(x|\alpha, \beta) \propto x^{\alpha-1}(1-x)^{\beta-1}$ , sampling from which is a standard procedure.

# References

- [1] B. Abraham and G. E. P. Box. Linear models and spurious observations. *Appl. Statist.*, 27:120–130, 1978.
- [2] B. Abraham and G. E. P. Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66(2):229–36, 1979.
- [3] A. N. Akansu and R. A. Haddad. *Multiresolution Signal Decomposition*. Academic Press, inc., 1992.
- [4] J. Allen. Short term spectral analysis, synthesis and modification by discrete Fourier transform. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-25:235–239, June 1977.
- [5] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.
- [6] J. B. Anderson and S. Mohan. Sequential coding algorithms: A survey and cost analysis. *IEEE Trans. Comms.*, COM-32:169–176, 1984.
- [7] D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36:99–102, 1974.
- [8] C. Andrieu, A. Doucet, and P. Duvaut. Bayesian estimation of filtered point processes using mcmc. In *in Proc. IEEE Asilomar Conf. on Signals, Systems and Computers IEEE, Asilomar, California*, November 1997.

- [9] K. Arakawa, D.H. Fender, H. Harashima, H. Miyakawa, and Y. Saitoh. Separation of a non-stationary component from the EEG by a nonlinear digital filter. *IEEE Trans. Biomedical Engineering*, 33(7):724–726, 1986.
- [10] P. E. Axon and H. Davies. A study of frequency fluctuations in sound recording and reproducing systems. *Proc. IEE, Part III*, page 65, Jan. 1949.
- [11] A. C. Bajpai, L. R. Mustoe, and D. Walker. *Advanced Engineering Mathematics*. Wiley, 1977.
- [12] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Chichester:Wiley, second edition, 1984.
- [13] R. J. Beckman and R. D. Cook. Outlier. . . . . s. *Technometrics*, 25(2):119–165, May 1983.
- [14] J. Berger, R. R. Coifman, and M. J. Goldberg. Removing noise from music using local trigonometric bases and wavelet packets. *J. Audio Eng. Soc.*, 42(10):808–818, October 1994.
- [15] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.
- [16] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by additive noise. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 208–211, 1979.
- [17] M. Berouti, R. Schwartz, and J. Makhoul. *Enhancement of speech corrupted by additive noise*, pages 69–73. In Lim [110], 1983.
- [18] S. A. Billings. Identification of nonlinear systems - A survey. *Proc. IEE, part D*, 127(6):272–285, November 1980.
- [19] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoustics, Speech and Signal Processing*, 27(2):113–120, April 1979.
- [20] S.F. Boll. Speech enhancement in the 1980's: Noise suppression with pattern matching. In S. Furui and M.M. Sondhi, editors, *Advances in Speech Signal Processing*, pages 309–325. Marcel Dekker, Inc., 1992.
- [21] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis, Forecasting and Control*. Prentice Hall, 3rd edition, 1994.
- [22] G. E. P. Box and G. C. Tiao. A Bayesian approach to some outlier problems. *Biometrika*, 55:119–129, 1968.

- [23] G. E. P. Box and G. C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, 1973.
- [24] K. Brandenburg. Perceptual coding of high quality digital audio. In Brandenburg and Kahrs [25], page 39.
- [25] K. Brandenburg and M. Kahrs, editors. *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Publishers, 1998.
- [26] P. Bratley, B. L. Fox, and E. L. Schrage. *A Guide to Simulation*. New York: Springer-Verlag, 1983.
- [27] C. N. Canagarajah. A single-input hearing aid based on auditory perceptual features to improve speech intelligibility in noise. *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State*, 1991.
- [28] C. N. Canagarajah. *Digital Signal Processing Techniques for Speech Enhancement in Hearing Aids*. PhD thesis, University of Cambridge, 1993.
- [29] O. Cappé. Enhancement of musical signals degraded by background noise using long-term behaviour of the short-term spectral components. *Proc. International Conference on Acoustics, Speech and Signal Processing*, I:217–220, 1993.
- [30] O. Cappé. *Techniques de réduction de bruit pour la restauration d'enregistrements musicaux*. PhD thesis, l'Ecole Nationale Supérieure des Télécommunications, 1993.
- [31] O. Cappé. Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor. *IEEE Trans. on Speech and Audio Processing*, 2(2):345–349, 1994.
- [32] O. Cappé and J. Laroche. Evaluation of short-time spectral attenuation techniques for the restoration of musical recordings. *IEEE Trans. on Speech and Audio Processing*, 3(1):84–93, January 1995.
- [33] B. P. Carlin, N. G. Polson, and D. S. Stoffer. A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of American Statistical Association*, 87(418):493–500, June 1992.
- [34] M. J. Carrey and I. Buckner. A system for reducing impulsive noise on gramophone reproduction equipment. *The Radio Electronic Engineer*, 50(7):331–336, July 1976.
- [35] C. K. Carter and R. Kohn. On Gibbs Sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

- [36] C.K. Carter and R. Kohn. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83:589–601, 1996.
- [37] C. K. Chui. *Wavelet Analysis and its Applications, Volume 1: An Introduction to Wavelets*. Academic Press, inc., 1992.
- [38] C. K. Chui. *Wavelet Analysis and its Applications, Volume 2: Wavelets: A Tutorial in Theory and Applications*. Academic Press, inc., 1992.
- [39] L. Cohen. Time-frequency distributions – a review. *Proc. IEEE*, 77(7), July 1989.
- [40] M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics – a comparative review. *Journal of American Statistical Association*, (434):94–883–904, 1996.
- [41] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, 1983.
- [42] P. de Jong and N. Shephard. The simulation smoother for time series models. *Biometrika*, 82(2):339–350, 1995.
- [43] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [44] M. Dendrinis, S. Bakamidis, and G. Carayannis. Speech enhancement from noise: A regenerative approach. *Speech Communication*, 10:45–57, 1991.
- [45] P. Depalle, G. García, and X. Rodet. Tracking of partials for additive sound synthesis using hidden Markov models. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 225–228. April 1993.
- [46] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
- [47] R. M. Dolby. An audio noise reduction system. *J. Audio Eng. Soc.*, 15(4):383–388, 1967.
- [48] J.L. Doob. *Stochastic processes*. John Wiley and Sons, 1953.
- [49] A. Doucet and P. Duvaut. Fully Bayesian analysis of hidden Markov models. In *in Proc. EUSIPCO, Trieste, Italy*, September 1996.
- [50] A. Doucet and P. Duvaut. Bayesian estimation of state space models applied to deconvolution of bernoulli-gaussian processes. *Signal Processing*, 57:147–161, 1997.

- [51] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [52] J. Durbin. Efficient estimation of parameters in moving-average models. *Biometrika*, 46:306–316, 1959.
- [53] A.J. Efron and H. Jeen. Pre-whitening for detection in correlated plus impulsive noise. *Proc. International Conference on Acoustics, Speech and Signal Processing*, II:469–472, 1992.
- [54] Y. Ephraim and D. Malah. Speech enhancement using optimal non-linear spectral amplitude estimation. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 1118–1121, Boston, 1983.
- [55] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-21(6):1109–1121, December 1984.
- [56] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Trans. Acoustics, Speech and Signal Processing*, 33(2):443–445, 1985.
- [57] Y. Ephraim and H. VanTrees. A signal subspace approach for speech enhancement. *Proc. International Conference on Acoustics, Speech and Signal Processing*, II:359–362, 1993.
- [58] W. Etter and G. S. Moschytz. Noise reduction by noise-adaptive spectral magnitude expansion. *J. Audio Eng. Soc.*, 42(5):341–349, 1994.
- [59] M. Feder, A. V. Oppenheim, and E. Weinstein. Maximum likelihood noise cancellation using the EM algorithm. *IEEE Trans. Acoustics, Speech and Signal Processing*, 37(2), February 1989.
- [60] G. D. Forney. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, March 1973.
- [61] S. Frühwirth-Schnatter. Data augmentation and dynamic linear models. *Journal of Time Series Analysis*, 15:183–202, 1994.
- [62] U. R. Furst. Periodic variations of pitch in sound reproduction by phonographs. *Proc. IRE*, page 887, November 1946.
- [63] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of American Statistical Association*, 85:398–409, 1990.

- [64] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [65] E. I. George and R. E. McCulloch. Variable selection via Gibbs sampling. *Journal of American Statistical Association*, 88(423):881–889, September 1993.
- [66] E.B. George and J.T. Smith. Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones. *J. Audio Eng. Soc.*, 40(6), June 1992.
- [67] E. N. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, pages 1253–1265, 1960.
- [68] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.
- [69] S. J. Godsill. Digital Signal Processing. U. K. Patent no. 2280763, 1993. *Patent Granted 1997*.
- [70] S. J. Godsill. *The Restoration of Degraded Audio Signals*. PhD thesis, Cambridge University Engineering Department, Cambridge, England, December 1993.
- [71] S. J. Godsill. Recursive restoration of pitch variation defects in musical recordings. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 233–236, Adelaide, April 1994.
- [72] S. J. Godsill. Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes. *International Statistical Review*, 65(1):1–21, 1997.
- [73] S. J. Godsill. Robust modelling of noisy ARMA signals. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, April 1997.
- [74] S. J. Godsill and A. C. Kokaram. Joint interpolation, motion and parameter estimation for image sequences with missing data. In *Proc. EUSIPCO*, September 1996.
- [75] S. J. Godsill and A. C. Kokaram. Restoration of image sequences using a causal spatio-temporal model. In *Proc. 17th Leeds Annual Statistics Research Workshop (The Art and Science of Bayesian Image Analysis)*, July 1997.



- [76] S. J. Godsill and P. J. W. Rayner. A Bayesian approach to the detection and correction of bursts of errors in audio signals. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 261–264, San Francisco, March 1992.
- [77] S. J. Godsill and P. J. W. Rayner. Frequency-domain interpolation of sampled signals. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 209–212, Mineapolis, April 1993.
- [78] S. J. Godsill and P. J. W. Rayner. The restoration of pitch variation defects in gramophone recordings. In *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State*, Mohonk, NY State, October 1993.
- [79] S. J. Godsill and P. J. W. Rayner. A Bayesian approach to the restoration of degraded audio signals. *IEEE Trans. on Speech and Audio Processing*, 3(4):267–278, July 1995.
- [80] S. J. Godsill and P. J. W. Rayner. Robust noise modelling with application to audio restoration. In *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State*, Mohonk, NY State, October 1995.
- [81] S. J. Godsill and P. J. W. Rayner. Robust noise reduction for speech and audio signals. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, May 1996.
- [82] S. J. Godsill and P. J. W. Rayner. Robust treatment of impulsive noise in speech and audio signals. In J.O. Berger, B. Petro, E. Moreno, L.R. Pericchi, F. Ruggeri, G. Salinetti, and L. Wasserman, editors, *Bayesian Robustness - proceedings of the workshop on Bayesian robustness, May 22-25, 1995, Rimini, Italy*, volume 29, pages 331–342. IMS Lecture Notes - Monograph Series, 1996.
- [83] S. J. Godsill and P. J. W. Rayner. Robust reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler. *IEEE Trans. on Speech and Audio Processing*, July 1998. Previously available as Tech. Report CUED/F-INFENG/TR.233.
- [84] S. J. Godsill, P. J. W. Rayner, and O. Cappé. Digital audio restoration. In K. Brandenburg and M. Kahrs, editors, *Applications of Digital Signal Processing to Audio and Acoustics*. Kluwer Academic Publishers, 1998.
- [85] S. J. Godsill and C. H. Tan. Removal of low frequency transient noise from old recordings using model-based signal separation techniques. In *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State*, Mohonk, NY State, October 1997.

- [86] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1989.
- [87] R.M. Gray and L.D. Davisson. *Random Processes: a Mathematical Approach for Engineers*. Prentice-Hall, 1986.
- [88] I. Guttman. Care and handling of univariate or multivariate outliers in detecting spuriousity – a Bayesian approach. *Technometrics*, 15(4):723–738, November 1973.
- [89] H. M. Hall. The generalized ‘t’ model for impulsive noise. *Proc. International Symposium on Info. Theory*, 1970.
- [90] J. P Den Hartog. *Mechanical Vibrations*. McGraw-Hill, 1956.
- [91] A.C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.
- [92] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.
- [93] S. Haykin. *Modern Filters*. MacMillan, 1989.
- [94] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, second edition, 1991.
- [95] P.J. Huber. *Robust Statistics*. Wiley and Sons, 1981.
- [96] A. J. E. M. Janssen, R. Veldhuis, and L. B. Vries. Adaptive interpolation of discrete-time signals that can be modeled as AR processes. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-34(2):317–330, April 1986.
- [97] H. Jeffreys. *Theory of Probability*. Oxford University Press, 1939.
- [98] N. L. Johnson and S. Kotz. *Distributions in Statistics: Continuous Univariate Distributions*. Wiley, 1970.
- [99] M. Kahrs. Digital audio system architecture. In Brandenburg and Kahrs [25], page 195.
- [100] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME J. Basic Eng.*, 82:35–45, 1960.
- [101] T. Kasparis and J. Lane. Adaptive scratch noise filtering. *IEEE Trans. Consumer Electronics*, 39(4), 1993.
- [102] G. R. Kinzie, Jr. and D. W. Gravereaux. Automatic detection of impulse noise. *J. Audio Eng. Soc.*, 21(3):331–336, April 1973.

- [103] G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series*. Lecture notes in statistics. Springer-Verlag, 1996.
- [104] B. Kleiner, R. D. Martin, and D. J. Thomson. Robust estimation of power spectra (with discussion). *Journal of the Royal Statistical Society, Series B*, 41(3):313–351, 1979.
- [105] A. C. Kokaram and S. J. Godsill. A system for reconstruction of missing data in image sequences using sampled 3D AR models and MRF motion priors. In *Computer Vision - ECCV '96*, volume II, pages 613–624. Springer Lecture Notes in Computer Science, April 1996.
- [106] A. C. Kokaram and S. J. Godsill. Detection, interpolation, motion and parameter estimation for image sequences with missing data. In *Proc. International Conference on Image Applications and Processing, Florence, Italy*, September 1997.
- [107] B. Koo, J. D. Gibson, and S. D. Gray. Filtering of coloured noise for speech enhancement and coding. *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 349–352, 1989.
- [108] L. Kuo and B. Mallick. Variable selection for regression models. *Sankhya*, 1997. (to appear).
- [109] R. Lagadec and D. Pelloni. Signal enhancement via digital signal processing. In *Preprints of the AES 74th Convention*, New York, 1983.
- [110] J. S. Lim, editor. *Speech Enhancement*. Prentice-Hall signal processing series. Prentice-Hall, 1983.
- [111] J. S. Lim and A. V. Oppenheim. All-pole modelling of degraded speech. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-26(3), June 1978.
- [112] J. S. Lim and A. V. Oppenheim. Enhancement and bandwidth compression of noisy speech. *Proc. IEEE*, 67(12), 1979.
- [113] J. Liu, W. H. Wong, and A. Kong. Covariance structure of the Gibbs sampler with applications to the comparison of estimators and augmentation schemes. *Biometrika*, 81:27–40, 1994.
- [114] L.J. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 1987.
- [115] G. B. Lockhart and D. J. Goodman. Reconstruction of missing speech packets by waveform substitution. *Signal Processing 3: Theories and Applications*, pages 357–360, 1986.

- [116] M. Lorber and R. Hoeldrich. A combined approach for broadband noise reduction. In *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State*, October 1997.
- [117] D. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, Calif. Inst. Tech., 1992.
- [118] R. C. Maher. A method for extrapolation of missing digital audio data. *95th AES Convention, New York*, 1993.
- [119] J. Makhoul. Linear prediction: A tutorial review. *Proc. IEEE*, 63(4):561–580, 1975.
- [120] S. Mallatt and W. L. Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Info. Theory*, pages 617–643, 1992.
- [121] Manouchehr-Kheradmandnia. *Aspects of Bayesian Threshold Autoregressive Modelling*. PhD thesis, University of Kent, 1991.
- [122] R. J. Marks II. *Introduction to Shannon Sampling and Interpolation Theory*. Springer-Verlag, 1991.
- [123] R.D. Martin. Robust estimation of autoregressive models. In *Directions in time series*, D.R. Brillinger and G.C. Tiao, eds., pages 228–254, 1980.
- [124] P. Martland. *Since Records Began: EMI, the First Hundred Years*. B.T. Battsford Ltd., 1997.
- [125] R. J. McAulay and M. L. Malpass. Speech enhancement using a soft-decision noise suppression filter. *IEEE Trans. Acoustics, Speech and Signal Processing*, 28(2):137–145, April 1980.
- [126] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-34(4):744–754, 1986.
- [127] R. J. McAulay and T. F. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. In W.B. Klein and K.K. Paliwal, editors, *Speech coding and synthesis*. Elsevier Science B.V., 1995.
- [128] R. E. McCulloch and R. S. Tsay. Bayesian inference and prediction for mean and variance shifts in autoregressive time series. *Journal of American Statistical Association*, 88(423):968–978, 1993.
- [129] R. E. McCulloch and R. S. Tsay. Bayesian analysis of autoregressive time series via the Gibbs sampler. *Journal of Time Series Analysis*, 15(2):235–250, 1994.

- [130] K. J. Mercer. *Identification of signal distortion models*. PhD thesis, University of Cambridge, 1993.
- [131] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.
- [132] N. J. Miller. Recovery of singing voice from noise by synthesis. *Thesis Tech. Rep.*, ID UTEC-CSC-74-013, Univ. Utah, May 1973.
- [133] E. Molloy. *High Fidelity Sound Reproduction*. Newnes, 1958.
- [134] S. Montresor. *Etude de la transformée en ondelettes dans le cadre de la restauration d'enregistrements anciens et de la détermination de la fréquence fondamentale de la parole*. PhD thesis, Université du Maine, Le Mans, 1991.
- [135] S. Montresor, J. C. Valière, and M. Baudry. Détection et suppression de bruits impulsionnels appliquées à la restauration d'enregistrements anciens. *Colloq. de Physique, C2, supplément au no. 2, Tome 51, Février*, pages 757–760, 1990.
- [136] T.K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazine*, pages 47–60, November 1996.
- [137] B. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, fourth edition, 1997.
- [138] J. A. Moorer and M. Berger. Linear-phase bandsplitting: Theory and applications. *J. Audio Eng. Soc.*, 34(3):143–152, 1986.
- [139] F. Mosteller and J.W. Tukey. *Data Analysis and Regression*. Addison-Wesley, Reading, Mass., 1977.
- [140] S.L. Ng. Estimation of corrupted samples in autoregressive moving average (ARMA) data sequences. Master's thesis, MEng dissertation, Cambridge University Engineering Dept., May 1997.
- [141] M. Niedźwiecki. Recursive algorithm for elimination of measurement noise and impulsive disturbances from ARMA signals. *Signal Processing VII: Theories and Applications*, pages 1289–1292, 1994.
- [142] M. Niedźwiecki and K. Cisowski. Adaptive scheme for elimination of broadband noise and impulsive disturbances from AR and ARMA signals. *IEEE Trans. on Signal Processing*, pages 528–537, 1996.
- [143] A. Nieminen, M. Miettinen, P. Heinonen, and Y. Nuevo. Music restoration using median type filters with adaptive filter substructures. In *Digital Signal Processing-87*, eds. V. Cappellini and A. Constantinides, North-Holland, 1987.

- [144] C. L. Nikias and M. Shao. *Signal Processing with  $\alpha$ -Stable Distributions and Applications*. John Wiley and Sons, 1995.
- [145] A.V. Oppenheim and R.W. Schaffer. *Digital Signal Processing*. Prentice-Hall, 1975.
- [146] J. J. K. Ó Ruanaidh and W. J. Fitzgerald. Interpolation of missing samples for audio restoration. *Electronic Letters*, 30(8), April 1994.
- [147] K. K. Paliwal and A. Basu. A speech enhancement method based on Kalman filtering. *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 177–180, 1987.
- [148] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, 3rd edition, 1991.
- [149] S. Park, G. Hillman, and R. Robles. A novel technique for real-time digital sample-rate converters with finite precision error analysis. *ICASSP*, 1991.
- [150] D. L. Phillips. A technique for numerical solution of certain integral equations of the first kind. *J. Ass. Comput. Mach.*, 9:97–101, 1962.
- [151] I. Pitas and A. N. Venetsanopoulos. *Nonlinear Digital Filters*. Kluwer Academic Publishers, 1990.
- [152] H. J. Platte and V. Rowedda. A burst error concealment method for digital audio tape application. *AES preprint*, 2201:1–16, 1985.
- [153] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. CUP, second edition, 1992.
- [154] R.D. Preuss. A frequency domain noise cancelling preprocessor for narrowband speech communications systems. *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 212–215, April 1979.
- [155] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1981.
- [156] J.G. Proakis, J.R. Deller, and J.H.L. Hansen. *Discrete-Time Processing of Speech Signals*. Macmillan, New York, 1993.
- [157] J.G. Proakis and D.G. Malonakis. *Introduction to Digital Signal Processing*. Macmillan, 1988.
- [158] T. F. Quatieri and R. J. McAulay. Audio signal processing based on sinusoidal analysis/synthesis. In Brandenburg and Kahrs [25], page 343.

- [159] L. R. Rabiner. Digital techniques for changing the sampling rate of a signal. *Digital Audio; collected papers from the AES premier conference*, pages 79–89, June 1982.
- [160] J. J. Rajan, P. J. W. Rayner, and S. J. Godsill. A Bayesian approach to parameter estimation and interpolation of time-varying autoregressive processes using the Gibbs sampler. *IEE Proc. Vision, image and signal processing*, 144(4), August 1997.
- [161] P. J. W. Rayner and S. J. Godsill. The detection and correction of artefacts in archived gramophone recordings. In *Proc. IEEE Workshop on Audio and Acoustics, Mohonk, NY State, Mohonk, NY State*, October 1991.
- [162] G. O. Roberts. Markov chain concepts related to sampling algorithms. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov chain Monte Carlo in practice*, pages 45–57. Chapman and Hall, 1996.
- [163] G.O. Roberts and S.K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society, Series B*, 59:291, 1997.
- [164] R.A. Roberts and C.T. Mullis. *Digital Signal Processing*. Addison-Wesley, 1997.
- [165] X. Rodet. Musical sound signal analysis/synthesis: sinusoidal+residual and elementary waveform models. In *IEEE UK Symposium on applications of Time-Frequency and Time-Scale Methods*, pages 111–120, August 1997.
- [166] J. J. K. Ó Ruanaidh. *Numerical Bayesian methods in signal processing*. PhD thesis, University of Cambridge, 1994.
- [167] J. J. K. Ó Ruanaidh and W. J. Fitzgerald. The restoration of digital audio recordings using the Gibbs' sampler. Technical Report CUED/F-INFENG/TR.134, Cambridge University Engineering Department, 1993.
- [168] J. J. K. Ó Ruanaidh and W. J. Fitzgerald. *Numerical Bayesian methods applied to signal processing*. Springer-Verlag, 1996.
- [169] X. Serra. *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. PhD thesis, Stanford University, October 1989.
- [170] N. Shephard. Partial non-Gaussian state space. *Biometrika*, 81(1):115–131, 1994.

- [171] T. G. Stockham, T. M. Cannon, and R. B. Ingebretsen. Blind deconvolution through digital signal processing. *Proc. IEEE*, 63(4):678–692, April 1975.
- [172] M. A. Tanner. *Tools for Statistical Inference, Second Edition*. Springer-Verlag, 1993.
- [173] C. W. Therrien. *Decision, Estimation and Classification*. Wiley, 1989.
- [174] C. W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, 1992.
- [175] L. Tierney. Markov chains for exploring posterior distributions (with discussion). *The Annals of Statistics*, (22):1701–1762, 1994.
- [176] H. Tong. *Non-linear Time Series*. Oxford Science Publications, 1990.
- [177] P. T. Troughton and S. J. Godsill. Bayesian model selection for time series using Markov chain Monte Carlo. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, April 1997.
- [178] P. T. Troughton and S. J. Godsill. A reversible jump sampler for autoregressive time series, employing full conditionals to achieve efficient model space moves. Technical Report CUED/F-INFENG/TR.304, Cambridge University Engineering Department, 1997.
- [179] P. T. Troughton and S. J. Godsill. Bayesian model selection for linear and non-linear time series using the Gibbs sampler. In John G. McWhirter, editor, *Mathematics in Signal Processing IV*. Oxford University Press, 1998.
- [180] P.T. Troughton and S.J. Godsill. MCMC methods for restoration of nonlinearly distorted autoregressive signals. In *Proc. European Conference on Signal Processing*, September 1998.
- [181] P.T. Troughton and S.J. Godsill. Restoration of nonlinearly distorted audio using Markov chain Monte Carlo methods. In *Preprint from the 104th Convention of the Audio Engineering Society*, May 1998.
- [182] P.T. Troughton and S.J. Godsill. A reversible jump sampler for autoregressive time series. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, April 1998.
- [183] D. Tsoukalas, M. Paraskevas, and J. Mourjopoulos. Speech enhancement using psychoacoustic criteria. *Proc. International Conference on Acoustics, Speech and Signal Processing*, II:359–362, 1993.



- [184] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1971.
- [185] J. C. Valière. *La Restauration d'Enregistrements Anciens par Traitement Numérique- Contribution à l'étude de Quelques techniques récentes*. PhD thesis, Université du Maine, Le Mans, 1991.
- [186] H. VanTrees. *Decision, Estimation and Modulation Theory, Part 1*. Wiley and Sons, 1968.
- [187] S. V. Vaseghi. *Algorithms for Restoration of Archived Gramophone Recordings*. PhD thesis, University of Cambridge, 1988.
- [188] S. V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction*. Wiley, 1996.
- [189] S. V. Vaseghi and R. Frayling-Cork. Restoration of old gramophone recordings. *J. Audio Eng. Soc.*, 40(10), October 1992.
- [190] S. V. Vaseghi and P. J. W. Rayner. A new application of adaptive filters for restoration of archived gramophone recordings. *Proc. International Conference on Acoustics, Speech and Signal Processing*, V:2548–2551, 1988.
- [191] S. V. Vaseghi and P. J. W. Rayner. Detection and suppression of impulsive noise in speech communication systems. *IEE Proceedings, Part 1*, 137(1):38–46, February 1990.
- [192] R. Veldhuis. *Restoration of Lost Samples in Digital Signals*. Prentice-Hall, 1990.
- [193] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Info. Theory*, IT-13:260–269, April 1967.
- [194] P.J. Walmsley, S.J. Godsill, and P.J.W. Rayner. Multidimensional optimisation of harmonic signals. In *Proc. European Conference on Signal Processing*, September 1998.
- [195] E. Weinstein, A. V. Oppenheim, M. Feder, and J. R. Buck. Iterative and sequential algorithms for multisensor signal enhancement. *IEEE Trans. on Signal Processing*, 42(4), April 1994.
- [196] M. West. Outlier models and prior distributions in Bayesian linear regression. *Journal of the Royal Statistical Society, Series B*, 46(3):431–439, 1984.
- [197] M. West. On scale mixtures of normal distributions. *Biometrika*, 74(3):99–102, 1987.

- [198] E. T. Whittaker. On a new method of graduation. *Proc. Edinburgh Math. Soc.*, 41:63–75, 1923.
- [199] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. MIT Press, 1949.

# Index

- $\alpha$ -stable noise, 238
- $n$ th order moment, 53
- $z$ -transform, 22–24
  - definition, 23
  - frequency response, 23
  - poles, 23
  - stability, 23
  - time shift theorem, 23
  - transfer function, 23
  - zeros, 23
- 45rpm, 4
- All-pole filter, 86
- Analogue de-clicking, 128
- Analogue restoration methods, 5
- Autocorrelation function, 64
- Autoregressive (AR) model, 86–89
  - approximate likelihood, 88, 196
  - backward prediction error, 132
  - conditional probability, 88
  - covariance estimate, 88
  - exact likelihood, 89, 157
  - excitation energy, 116
  - Gaussian excitation, 88
  - observed in noise, 90
  - separation of two AR processes, 156–157
  - state-space form, 90
  - statistical modelling and estimation, 87
- Autoregressive moving average (ARMA) model, 86
- Axioms of probability, 41
- Background noise, 6
- Background noise reduction, *see* Hiss reduction
- Band-pass filter, 67
- Bayes factor, 80
- Bayes risk, 75
- Bayes rule, 43, 47
  - cumulative distribution function (CDF), 49
  - inferential procedures, 47
  - probability density function, 50
  - probability mass function (PMF), 48
- Bayesian decision theory, 79–85

- Bayesian estimation, 73–79
- Bayesian regulariser, 178
- Bell, Alexander Graham, 4
- Berliner, Emile, 4
- Bernoulli model, 101, 211
- Beta distribution, 304
- Blumlein, Alan, 4
- Breakage, 158
- Breakages, 6, 153
- Broadband noise reduction, *see* Hiss reduction
- Burst noise, 192
- Caruso, 5
- CEDAR Audio, 16
- Central moment, 53
- Change-points, 234
- Characteristic function, 53
  - inverse, 54
  - moments of a random variable, 54
- Chi-squared, 56
- Cholesky decomposition, 116
- Classification error probability, 80
- Click detection, *see* Detection of clicks
- Click removal, 99–134
  - Gibbs sampler, 241
  - implementation/results, 248
- Clicks, 6
- Compact disc (CD), 5, 16
- Conditional distribution, 47
  - cumulative distribution function (CDF), 49
  - probability density function (PDF), 49
  - probability mass function (PMF), 48
- Conditional probability, 42
- Convolution, 20–22
  - sum of random variables, 54
- Convolution of PDFs, 51
- Cost function
  - Bayesian, 75
  - quadratic, 76
  - uniform, 76
- Cost functions
  - Bayesian, 77
- Covariance matrix, 60
- Crackles, 6
- Cross-correlation function, 65
- Cumulative distribution function (CDF), 45, 46
- Data windows
  - see* Windows, 175
- Derived distribution, 55
- Derived PDF, 58
- Detection of clicks, 101, 127–133
  - i.i.d. Gaussian noise assumption, 198
  - adaptations to AR detector, 131–132
  - analysis and limitations, 130–131
  - ARMA model, 133
  - autoregressive (AR), 128
  - Bayesian, 191–204
    - autoregressive (AR), 195
    - complexity, 201
    - loss functions, 200
    - marginalised, 201
    - relationship with LS interpolator, 200
    - relationship with simple AR, 203
    - results, 215–231
    - selection of optimal state, 200
  - Bayesian sequential
    - complexity, 210
    - Kalman filter implementation, 210
    - state culling, 207
    - state culling strategy, 212
    - update equations, 209
  - false Alarms, 101
  - false detection, 131
  - Gibbs sampler, 244
  - high-pass filter, 128

- likelihood for Gaussian AR
  - data, 197
- Markov chain model, 192
- matched filter, 132
- matched filter detector, 132–133
- missed detection, 101
- model-based, 128
- multi-channel, 203
- noise generator prior, 194, 210
- noisy data case, 193
- noisy data model, 202
- non-standard Bayesian loss functions, 213
- probability of error, 127
- probability ratio test, 199
- Recursive update for posterior probability, 207
- sequential Bayesian, 203, 205–214
- sinusoid+AR residual model, 133
- statistical methods, 134
- switched noise model, 194
- threshold selection, 101, 131
- uniform prior, 194
- Deterministic signals, 38
- Differential mappings, 55
- Digital audio tape (DAT), 16
- Digital filters, 24–25
  - finite-impulse-response (FIR), 25
  - infinite-impulse-response (IIR), 24
- Digital re-sampling, 173
- Discrete Fourier transform (DFT), 25–27, 34, 126, 136, 175
  - inverse, 26
- Discrete probability space, 41
- Discrete time Fourier transform (DTFT), 19–20, 66
- Distortion
  - peak-related, 99
- Edison, Thomas, 4
- Electrical noise, 135
- Elementary outcomes, 40
- Ensemble, 61
- Equalisation, 135
- Event based probability, 39
  - fundamental results, 42
- Event space, 40
- Evidence, 80
  - linear model, 81
- Expectation, 52–54
  - function of random variable, 52
  - linear operator, 52
  - random variable, 52
- Expectation-maximisation (EM), 92–93, 181, 234
- Expected risk, 80
- Fast Fourier transform (FFT), 26, 34–38, 136
  - butterfly structure, 35
  - in-place calculation, 37
- Finite-impulse-response (FIR) filters, 25
- Flutter, 6, *see* Pitch variation defects
- Fourier series, 17
- Fourier transform
  - relation with characteristic function, 53
- Frequency modulation, 173
- Frequency response, 22, 23
- Frequency shift theorem, 17
- Frequency tracking, 173
  - birth and death of components, 183
- Fully Bayesian restoration, 233–259
- Functions of random variables, 54
- Functions of random vectors, 58
- Gamma density, 277
- Gamma function, 277
- Gamma integral, 277
- Gaussian, 55, 59

- multivariate, 59, 275
  - multivariate integral, 276
  - univariate, 275
- Generalised Gaussian, 238
- Gibbs sampler, 94, 119, 234
- Global defects, 6
- Global degradation, 6
- Gramophone discs, 1
  - fluctuation in turntable speed, 172
  - imperfectly punched centre hole, 171
- Hamming window, 32, 136
- Hanning window, 29, 32, 136
- Heavy-tailed noise distribution, 233
- High-pass filter
  - removal of low frequency noise pulses, 154
- Hiss reduction, 135–149
  - model based, 148–149
  - noise artefacts, 141
  - Spectral domain
    - psychoacoustical methods, 148
  - spectral domain, 136–148
    - maximum likelihood (ML), 148
    - minimum mean-squared error (MMSE), 148
  - sub-space methods, 148
  - wavelets, 148
  - Wiener, 138
- Impulse train, 17
- Impulsive stimuli, 154
- Independence, 44
- Infinite-impulse-response (IIR) filters, 24
- Innovation, 71
- Interpolation, 101–127
  - autoregressive (AR), 106–122
    - adaptations to, 110
    - AR plus basis function representation, 111
    - examples, 108
    - incorporating a noise model, 119
    - least squares (LS), 106–107
    - maximum *a posteriori* (MAP), 107–108
    - pitch-based adaptation, 111
    - random sampling methods, 116–119
    - sequential methods, 119–122
    - sinusoid plus AR, 115–116
    - unknown parameters, 108–110
  - autoregressive moving-average (ARMA), 122–126
  - autoregressive (AR)
    - AR plus basis function representation, 116
  - expectation-maximisation (EM), 239
  - frequency domain, 126–127
  - Gaussian signals, 103–105
    - incorporating a noise model, 105
  - Gibbs sampler, 242
  - model-based, 102
  - transform domain methods, 126
- Interpolation of corrupted samples, *see* Interpolation
- Inverse image method, 54
- Inverted-gamma density, 277
- Inverted-gamma prior, 238
- Jacobian, 59, 178
- Jeffreys prior, 238, 278, 286
- Jeffreys' prior, 79
- Joint PDF, 50
- Kalman filter, 90, 91, 125, 205
  - evaluation of likelihood, 91
  - prediction error decomposition, 91
- Laplace transform, 22

- Least squares (LS)
  - relationship with maximum likelihood, 73
- Likelihood function, 71
- Linear model, general, 70
  - MAP estimator, 77
- Linear prediction, 86
- Linear time invariant (LTI), 20
- Localised defects, 6
- Localised degradation, 6
- Location parameter, 79
- Loss function, 80
  - symmetric, 80
- Low frequency noise pulses, 153–170
  - cross-correlation detector, 156
  - detection, 156
  - experimental results, 162
  - Kalman filter implementation, 163
  - matched filter detection, 156
  - template method, 153, 154
  - unknown parameter estimation, 160
- Low-pass filter, 19
- LP (vinyl), 5
- Magnetic tapes, 1, 4, 135
  - unevenly stretched, 171
- MAP classification, 195
- Marginal density, 119
- Marginal probability, 50, 57, 179
- Marginalisation, 77–78
- Markov chain, 192
  - noise generator model, 206
  - noise generator process, 211
- Markov chain Monte Carlo (MCMC), 93–95
  - review of methods, 234
- Masking
  - simultaneous, 148
- Masking of small clicks, 133
- Matched filter, 132
- Maximum likelihood
  - general linear model, 73
- Maximum likelihood (ML), 71–73, 102
  - ill-posed, 178
- Maximum *a posteriori* (MAP), 102, 127
- Maximum *a posteriori* (MAP) estimator, 76
- Median filter, 102
- Minimum mean-squared error
  - Kalman filter, 91
- Minimum mean-squared error (MMSE), 76, 102
- Minimum variance estimation, 103
- Missing data, 77
- Modelling of clicks, 100–101
- Modelling of signals, 85–92
- Multiresolution methods, 128
- Musical noise, 141
  - elimination, 145
  - masking, 147
- Musical signals
  - fundamental and overtones, 173
  - note slides, 173
  - tracking of tonal components, 175
- Noise
  - additive, 100
  - bursts, 100
  - clicks, 99
  - clustering, 100
  - localised, 99, 100
  - replacement, 100
- Noise artefacts, 141
- Noise reduction, *see* Hiss reduction
- Non-Gaussian noise, 234
- Non-linear distortions, 6
- Non-linear models
  - stiffening spring, 158
- Non-uniform sampling, 185
- Non-uniform time-warping, 172
- Normal distribution, *see* Gaussian
- Nyquist frequency, 190

- Nyquist sampling theorem, 16–19
- Observation equation, 90
- Optical sound tracks, 153
- Outliers, 127, 191, 234
- Overlap-add synthesis, 138
- Parameter estimation, 70–79
- Partials, *see* Musical signals
- Phase
  - sensitivity of ear, 140
- Phonautograph, 2
- Phonograph, 4
- Pitch period, 102, 111
- Pitch variation defects, 171–190
  - additive model, 176
  - AR model, 180
  - Bayesian estimator, 178
  - deterministic models, 182
  - discontinuities in frequency tracks, 182
  - experimental results, 183
  - frequency tracking, 173
  - generation of pitch variation curve, 176
  - log-additive model, 176
  - multiplicative model, 176
  - prior models, 180
  - smoothness prior, 179, 181
- Point estimates, 75
- Pops, *see* Low frequency noise pulses
- Posterior probability, 43, 74
  - recursive update, 205
- Power spectrum, 66
- Power subtraction, 140
- Pre-whitening filter, 132
- Prediction error decomposition, 91
- Prior distribution
  - choice of, 78
  - conjugate, 79
  - Gaussian, 78
  - inverted-gamma, 79
  - non-informative, 79
- Prior probability, 43
- Probability
  - frequency-based interpretation, 40
- Probability density function (PDF), 45, 46
- Probability mass function (PMF), 45, 46
- Probability measure, 41
- Probability spaces, 40
- Probability theory, 39–60
- Quasi-periodic signals, 127
- Random experiment, 40
- Random periodic process, 127
- Random process, 60–67
  - autocorrelation function, 64
  - continuous time, 64
  - cross-correlation function, 65
  - definition, 64
  - discrete time, 64
  - linear system response, 67
  - mean value, 64
  - power spectrum, 66
  - stationarity, 65
  - strict-sense stationarity, 65
  - wide-sense stationarity, 66
- Random signals, 38, 61
- Random variables (RVs), 44–56
  - continuous, 45
  - definition, 45
  - discrete, 45
  - probability distribution, 45
- Random vectors, 56–60
  - Bayes rule, 57
  - CDF, 56
  - conditional distribution, 57
  - PDF, 57
- Real time processing, 15, 16, 133
- Rectangular window, 32
- Recursive least squares (RLS), 84
  - relationship with Kalman filter, 92
- Residual noise, 141
- Restoration
  - statistical methods, 133



- Risk function, 213
- Sample rate conversion, 190
  - time-varying, 173
  - truncated sinc function, 190
- Sample space, 40
- Scale mixture of Gaussians, 233, 238
- Scale parameter, 77, 79
- Scratches, 6, 153
- Second difference smoothness model, 162
- Separation of AR processes, *see* Autoregressive (AR) model
- Sequential Bayes, *see* Click detection
- Sequential Bayesian classification, 82–85
  - Linear model, 85
- Sigma algebra, 40
- Sinusoidal model, 71, 126
  - coding of speech and audio, 175
- Smearing, 29
- Smoothness prior, 181
- Sonic Solutions, 16
- Sound recordings
  - film sound tracks, 99
- Source-filter model, 86
- Spectral leakage, 29
- Spectral subtraction, 140
- State update equation, 90
- State-space model, 90–92
  - non-Gaussian, 234
  - non-linear, 234
- Stationarity, 65
  - short-term, 128
- Step-like stimuli, 154
  - mechanical response, 158
- Stiffening spring mechanical system, 158
- Stochastic processes, *see* random processes
- Strict-sense stationarity, 65
- Student-t distribution, 238
- Sum of random variables, 51, 54
- Switched noise model, 194
- Thumps, *see* Low frequency noise pulses
- Time series, 64
- Tone arm resonance, 158
- Total probability, 43
  - PDF, 51
  - probability mass function (PMF), 48
- Transfer function, 23
- Transient noise pulses, *see* Low frequency noise pulses
- Tremolo, 176, *see* Pitch variation defects
- Uniform prior, 179, 194, 238, 278
- Uniform sampling, 64
- Variance, 53
- Vibrato, 176, *see* Pitch variation defects
- Viterbi algorithm, 212
- Volterra model, 158
- Wavelets, 126, 128
- Wax cylinders, 1, 4, 135
- Wide-sense stationarity, 66
- Wiener, 135
- Wiener filter, 138
- Windows, 27–32
  - Bartlett, 32
  - continuous signals, 27
  - discrete-time signals, 31
  - frequency smearing, 29
  - generalised Hamming window, 32
  - Hamming, 32, 175
  - Hanning, 32
  - Hanning window, 29
  - Kaiser, 32
  - Parzen, 32
  - rectangular, 32
  - spectral leakage, 29

Tukey, 32

Wow, 6, *see* Pitch variation defects