Simulation Methods for Linear and Nonlinear Time Series Models with Application to Distorted Audio Signals

Simulation Methods for Linear and Nonlinear Time Series Models with Application to Distorted Audio Signals

Paul Thomas Troughton

Clare College

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

To my parents

Declaration

I hereby certify that, except as indicated in the text, the work described in this dissertation is entirely original. It is not the result of work done in collaboration, and has not been submitted to any other University. This dissertation contains 38 figures and less than 35 000 words.

<div align="right">

Paul Troughton

</div>

Acknowledgements

Keywords

The following keywords may be useful for indexing purposes:

Audio restoration; Bayesian time series modelling; Gibbs sampler; Markov chain Monte Carlo; Model selection; Noise reduction; Nonlinear autoregressive modelling; Nonlinear model estimation; Quantisation distortion; Reversible-jump MCMC; Subset selection; Truncated Gaussian distributions; Volterra polynomial expansion.

Summary

This dissertation is concerned with the development of Markov chain Monte Carlo (MCMC) methods for the Bayesian restoration of degraded audio signals. First, the Bayesian approach to time series modelling is reviewed, then established MCMC methods are introduced.

The first problem to be addressed is that of model order uncertainty. A reversible-jump sampler is proposed which can move between models of different order. It is shown that faster convergence can be achieved by exploiting the analytic structure of the time series model.

This approach to model order uncertainty is applied to the problem of noise reduction using the simulation smoother. The effects of incorrect autoregressive (AR) model orders are demonstrated, and a mixed model order MCMC noise reduction scheme is developed.

Nonlinear time series models are surveyed, and the advantages of linear-in-the-parameters models explained. A nonlinear AR (NAR) model, based on the Volterra polynomial expansion, is described, in which the model selection problem becomes one of subset selection. Subset selection methods are reviewed, including Bayesian MCMC methods. A new MCMC approach is formulated, using latent indicator variables in a Gibbs sampler. It is shown that using analytic results to create a multi-move sampler leads to better performance.

The effects, and some sources, of distortion in audio recordings are described. The few previous attempts to remove these types of distortion are reviewed. A general method is proposed, based on a cascade model in which the signal is modelled as an AR process, and the nonlinear channel as an NAR process. The model structure, order and parameters are jointly estimated in a MCMC scheme. The method is extended to process long sequences, in which the audio signal cannot be modelled as stationary, by estimating the nonlinear model structure and parameters jointly across all the blocks.

The quantisation distortion present in limited word length digital audio is examined. A model-based framework is proposed for restoring such quantised signals. In order to implement this, methods are investigated for drawing samples from truncated multivariate Gaussian distributions. The restoration is improved by the use of sinusoidal modelling with AR residuals.

Contents

List of figures

List of tables

List of algorithms

Abbreviations & notation

i.i.d.	independent and identically distributed
r.m.s.	root mean square
c.d.f.	cumulative distribution function
p.d.f.	probability density function
$\mathbf{a}^{(k)}$	vectors of parameters associated with model of order k
$\mathbf{b}, \mathbf{c}$	vectors of parameters
$\boldsymbol{\beta}, \boldsymbol{\gamma}$	vectors of binary indicator variables associated with $\mathbf{b}, \mathbf{c}$
θ	scalar
$\boldsymbol{\theta}$	column vector
n_θ	number of elements in $\boldsymbol{\theta}$
θ_i	ith element of $\boldsymbol{\theta}$
$\boldsymbol{\theta}_{[-i]}$	all but the ith element of $\boldsymbol{\theta}$
$\boldsymbol{\theta}_{\langle i...j \rangle}$	the ith to jth elements of $\boldsymbol{\theta}$
$\boldsymbol{\theta}_u$	the elements of $\boldsymbol{\theta}$ which are being *updated*
$\boldsymbol{\theta}_f$	the elements of $\boldsymbol{\theta}$ which are remaining *fixed*
$\boldsymbol{\theta}_\beta$	the elements of $\boldsymbol{\theta}$ which correspond to ones in $\boldsymbol{\beta}$
n_β	the number of ones in $\boldsymbol{\beta}$
$\mathbf{0}$	column vector of zeros
$\mathbf{I}_k$	k-dimensional identity matrix
$\mathbf{Q}$	matrix
$\mathbf{Q}^T$	transpose of $\mathbf{Q}$
$\mathbf{Q}^{-1}$	inverse of $\mathbf{Q}$
$|\mathbf{Q}|$	determinant of $\mathbf{Q}$
$p(\boldsymbol{\theta})$	joint p.d.f. for the elements of $\boldsymbol{\theta}$
$p(\boldsymbol{\theta} \mid \phi)$	p.d.f. for $\boldsymbol{\theta}$, conditioned on ϕ
$\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$	$\boldsymbol{\theta}$ is drawn randomly from the distribution $p(\boldsymbol{\theta})$
$\mathrm{N}\left(\theta \mid \mu, \sigma^2\right)$	univariate Gaussian distribution in θ with mean μ and variance σ^2
$\mathrm{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \mathbf{C})$	multivariate Gaussian distribution in $\boldsymbol{\theta}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$
$\mathrm{IG}(\theta \mid \alpha, \beta)$	inverse Gamma distribution in θ with parameters α, β
$\mathrm{U}\left(\theta \mid b_-, b_+\right)$	uniform distribution in θ, bounded between b_- and b_+

Introduction

<div style="text-align:right">1</div>

1.1 Overview

In the last decade, methods for restoring damaged audio recordings have progressed from *ad hoc* methods, motivated primarily by ease of implementation, towards a more sophisticated approach based on mathematical modelling of the signal and degradation processes.

This thesis addresses quantisation distortion, a previously unresearched audio restoration problem arising in poorly manipulated digital audio signals; develops a new approach to the restoration of nonlinearly distorted audio; and improves an existing model-based noise reduction algorithm. Each of these problems is treated as a Bayesian estimation task.

The Bayesian methodology provides an elegant and consistent approach to statistical problems, in which all assumptions are explicit. Unfortunately, it tends to require the evaluation of high-dimensional integrals. Until recently, the use of Bayesian methods has been confined by the need to design the problem carefully so that the integrals are of a form which can be solved analytically or for which there is a good approximation; conventional numerical integration techniques are of limited value in such problems.

Markov chain Monte Carlo (MCMC) methods, which were first used in the 1950s, have recently been rediscovered by the Bayesian statistical community as a means to perform these integrals. This has allowed a much wider range of problems to be addressed in a Bayesian framework, with the emphasis shifting towards sensible modelling of the actual problem, rather than forcing the problem to fit a convenient model.

Digitised audio signals consist of vast quantities of data—a compact disc contains over five million samples per minute of recording—so research into audio restoration techniques has tended to concentrate on schemes which require little computation.

This thesis instead uses a fully Bayesian approach, and uses MCMC for implementation. This can be highly computationally demanding, but with the rapid, and continuing, rise in available computing power,[1] approaches which would have been intolerably slow just a few years ago are now merely inconvenient, and should work at a useful speed within a few years. Taking such an approach allows previously intractable problems to be addressed.

But the search for speed is not entirely abandoned: by using those analytical results which are available for the models under consideration, more efficient MCMC model and subset selection techniques are developed. Another contribution is a new method for drawing samples from truncated multivariate Gaussian distributions, which also lessens the required computation.

1.2   Structure of thesis

Chapter 2 introduces the Bayesian approach to time series modelling, together with a variety of linear and nonlinear time series models. Chapter 3 reviews Markov chain Monte Carlo methods, which can be applied to otherwise intractable Bayesian problems.

Chapter 4 examines the problem of model order uncertainty, reviews previous MCMC approaches, then develops a method, based on reversible-jump MCMC, which exploits some analytic properties of autoregressive models. Some of this work has previously been published in [188, 192]. This technique is then applied to the problem of noise reduction in audio, where the correct choice of model order is found to be crucial.

Chapter 5 considers model selection in a situation typical of nonlinear modelling: there are many candidate model terms, from which a small subset must be chosen. Some of this work has previously been published in [187, 189] and submitted as [193].

In Chapter 6, the subset selection method is used to model a nonlinear channel in order to restore audio signals which have passed through it. The audio signals are modelled using the approach developed in Chapter 4. Treating the audio signal as piecewise stationary and the channel as

---

[1] Gordon Moore, the co-founder of Intel, predicted in 1965 that the gate density (and hence processing power) of microprocessors would double and the cost halve every 24 months. In fact this has happened roughly every 18 months since then.

time-invariant introduces great flexibility. This work has previously been presented in [190, 191] and submitted as [193].

Chapter 7 addresses a different type of nonlinear distortion which affects audio: quantisation distortion. A Bayesian restoration method is proposed, based on an autoregressive model. Implementing this requires samples to be drawn from bounded multivariate Gaussian distributions, which is non-trivial. To improve performance, a sinusoidal model is introduced, with an autoregressive model used for the residuals. Parts of this work have been presented in [194] and will be presented in [186, 195].

Finally, Chapter 8 presents conclusions and suggests possible directions for future research.

Bayesian time series modelling 2

2.1 Bayesian paradigm

Statisticians are concerned with inferring facts from limited amounts of available data. A huge variety of techniques has been developed in classical statistics for distinguishing significant phenomena from artefacts resulting from random variation. Different techniques make different implicit assumptions, such as Gaussianity, about the problem.

The Bayesian approach [24, 107], is one alternative, based on probability theory, which allows much greater flexibility in inference while remaining consistent. Rather than having to decide which classical significance test is most appropriate to the problem, in Bayesian analysis the *posterior distribution* is always used.

The basic Bayesian approach is to set up a model for the system, leading to a *joint probability distribution* for all the known and unknown variables in the system. Nuisance parameters—those whose values are not of interest—can then be removed through marginalisation. The marginal posterior distribution for the quantities of interest is then used for inference, either by finding its peak, or by finding the expected value of some function by integrating over it.

Partly because of the difficulty in evaluating the necessary integrals and maximisations, Bayesian methods were not widely used for the first two hundred years after they were first mooted by the Rev. Thomas Bayes in 1763 [15]. The continuing increase in available computing power since the 1970s has, however, made numerical evaluation feasible and led to their much wider application.

2.1.1   Defining "probability"

The definition of probability taught in schools and used in most dictionary definitions is the *frequentist* definition, based on the relative frequency of occurrence of different outcomes in repeated trials of some experiment. An alternative is *subjective* probability, which is a measure of the plausibility of a proposition, conditional on the observer's knowledge.

We denote the probability of an event A, conditional on our knowledge that event B has occurred, as $\Pr(A \mid B)$. Where we are considering continuous random variables, say G and H, rather than discrete events, we use the *probability distribution*, $p_{G|H}(g \mid h)$, such that

$$\Pr(g_1 < G < g_2 \mid H = h) = \int_{g_1}^{g_2} p_{G|H}(g \mid h) \, dg \qquad (2.1)$$

The probability distribution can be represented in functional form as a *probability density function* (p.d.f.).

To simplify notation, we will not usually make the distinction between random variables, such as G, and their values, such as g. We also will not label probability distributions using subscripts where the meaning is clear from the variables or the context.

For a complete discussion of probability and the Bayesian approach, see Gelman, Carlin, Stern & Rubin [62, Ch. 1] and Bernardo & Smith [19, Ch. 1–2].

2.1.2   Bayes' theorem

Bayes' theorem is a simple relationship of conditional probabilities:

$$\Pr(B \mid A, C) = \frac{\Pr(A \mid B, C) \Pr(B \mid C)}{\Pr(A \mid C)} \qquad (2.2)$$

which can be derived straightforwardly from the product rule for probabilities, which itself follows naturally from Aristotelian inductive reasoning [106, Ch. 1–2].

The keys to the Bayesian approach are the treatment of unknown parameters as random variables[1], and the interpretation of Bayes' theorem as

---

[1]or, possibly, constants about which we have imperfect knowledge, which we express as a probability distribution [151, §3.2.1]

a model of the learning process:

$$p(\boldsymbol{\theta} \mid \mathbf{x}, \mathcal{M}) = \frac{p(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta} \mid \mathcal{M})}{p(\mathbf{x} \mid \mathcal{M})} \qquad (2.3)$$

where

- $p(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M})$ is the *likelihood*, which is a function of $\boldsymbol{\theta}$, expressing the probability distribution of the data $\mathbf{x}$ conditional on the value of the parameters $\boldsymbol{\theta}$ and any underlying modelling assumptions, $\mathcal{M}$;

- $p(\boldsymbol{\theta} \mid \mathcal{M})$ is the *prior*, which expresses our knowledge (or lack thereof) of the parameter values before examining the data;

- $p(\mathbf{x} \mid \mathcal{M})$ is the *evidence*, which can be regarded as a normalising constant; and

- $p(\boldsymbol{\theta} \mid \mathbf{x}, \mathcal{M})$ is the *posterior*, which is what we know about the parameters after examining the data.

For an illustration of how counterintuitive correct inference can be, see [9]. To clarify notation, from now on any conditioning on modelling assumptions, $\mathcal{M}$, will be omitted except where relevant.

2.1.3 Priors

The prior distribution represents belief about (or knowledge of) the parameter values before examining the data. As shown in Figure 2.1, a narrow prior distribution implies precise knowledge, and will dominate the posterior. On the other hand, if the prior distribution is broad, it conveys little knowledge; if it overlaps with, and is much broader than, the likelihood, it will have little influence on the posterior.

2.1.3.1 *Non-informative priors*

Jeffreys [107] describes how to choose "non-informative" prior distributions, which express complete ignorance of the parameter values, and hence do not influence the posterior. For location parameters (such as means or offsets), these are uniform, whereas for scale parameters (such as noise variances), they take the form $p(\sigma^2) \propto \frac{1}{\sigma^2}$. These are known as *Jeffreys' priors*.

(a) *Broad prior*   (b) *"Informative" prior*
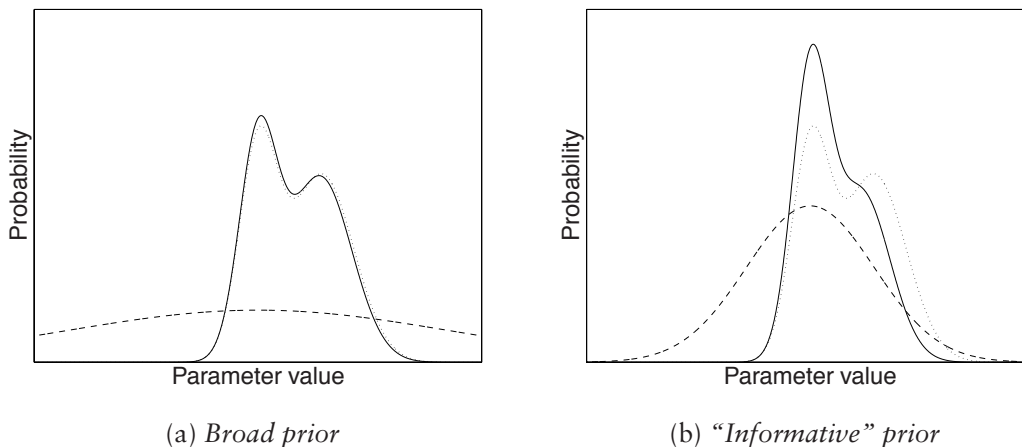
Figure 2.1. Effect of priors: *Probability distributions of the prior* (dashed), *likelihood* (dotted), *and resulting posterior* (solid). *It can be seen that whereas in* (a), *the broad prior has very little effect on the posterior, the prior in* (b) *is sufficiently informative to make the posterior unimodal.*

In many cases, there is vague prior knowledge of the range in which the parameter values are likely to lie. Priors can then be chosen which are equal to the Jeffreys' priors within that range but have zero probability outside. Even if such knowledge is unavailable, rough bounds can be estimated from the data, and used to form an *ignorance prior* [3] which is non-informative across the relevant range.

Unlike unbounded Jeffreys' priors, these bounded distributions are *proper*, *i.e.*

$$\int p(\boldsymbol{\theta}) \, d\boldsymbol{\theta} < \infty \qquad (2.4)$$

and can hence be *normalised*, so that they integrate to unity. This ensures that the evidence, and hence the posterior (eq. 2.3), are also proper so that they can be used for drawing samples, estimating values, or comparing models.

A more flexible approach, in the absence of definite prior information, is to choose a prior distribution which is close to non-informative within the range of interest, but has a convenient, proper form, such as a conjugate prior.

2.1.3.2 Conjugate priors

For a given likelihood function, a *conjugate prior* [19] is one which will lead to the posterior taking the same form as the prior, which can greatly simplify computation, particularly in sequential learning.

For example, if the data, $\mathbf{x}$, is assumed to be i.i.d. Gaussian with mean μ_x and variance σ_x^2, and we wish to estimate μ_x, the Gaussian likelihood function can be transformed, using the identity described in §A.2, to be a Gaussian in μ_x:

$$p(\mathbf{x} \mid \mu_x, \sigma_x^2) \propto \prod_{i=1}^{n_x} \mathrm{N}(x_i \mid \mu_x, \sigma_x^2) \tag{2.5}$$

$$\propto \mathrm{N}(\mu_x \mid \mu_{\text{likelihood}}, \sigma_{\text{likelihood}}^2) \tag{2.6}$$

where $\mu_{\text{likelihood}}$ is the sample mean of $\mathbf{x}$, $\sigma_{\text{likelihood}}^2 = \frac{\sigma_x^2}{n_x}$, and n_x is the length of the data vector. If we assume a Gaussian prior for μ_x:

$$p(\mu_x) = \mathrm{N}(\mu_x \mid \mu_{\text{prior}}, \sigma_{\text{prior}}^2) \tag{2.7}$$

then the posterior can be derived using Bayes' theorem (eq. 2.3) as

$$p(\mu_x \mid \mathbf{x}, \sigma_x^2) \propto p(\mathbf{x} \mid \boldsymbol{\theta}) \times p(\boldsymbol{\theta}) \tag{2.8}$$

$$\propto \mathrm{N}(\mu_x \mid \mu_{\text{likelihood}}, \sigma_{\text{likelihood}}^2) \times \mathrm{N}(\mu_x \mid \mu_{\text{prior}}, \sigma_{\text{prior}}^2) \tag{2.9}$$

$$\propto \mathrm{N}(\mu_x \mid \mu_{\text{posterior}}, \sigma_{\text{posterior}}^2) \tag{2.10}$$

since the product of two Gaussians is itself Gaussian (§A.1). The parameters of the posterior Gaussian distribution, $\mu_{\text{posterior}}$ and $\sigma_{\text{posterior}}^2$, can be derived using the results of §A.1, which deals with the general multivariate case. Similarly, if an inverse Gamma prior were used for the variance, σ_x^2, then the posterior, $p(\sigma_x^2 \mid \mathbf{x}, \mu_x)$, would also be inverse Gamma [19, §A.2].

2.1.3.3 Hyperparameters

Having chosen a functional form for a prior distribution, we then choose values for the *hyperparameters*, *i.e.* the parameters of the prior distribution, which best represent our prior knowledge.

Hierarchical modelling [62, Ch. 5] is an alternative approach, in which the hyperparameters are themselves treated as unknowns, with our limited

knowledge of them expressed by *hyperpriors*.

In §2.1.3.2, separate priors were assigned to μ_x and σ_x^2. If prior knowledge suggested it, they could be assigned dependent priors. If the nature of their interaction, *i.e.* the joint distribution, is not known precisely, the limited knowledge could be expressed as a hyperprior, the parameters of which—*hyperparameters*—can then either be estimated, if they are of interest, or integrated out if they are not (see §2.1.5). For complex systems, *directed graphs* can be used to represent the dependencies between the parameters (see *e.g.* [134]).

2.1.4 Parameter estimation

After the likelihood function for the chosen model has been derived, and a prior distribution has been constructed for $\boldsymbol{\theta}$, incorporating any *a priori* knowledge, the posterior distribution, $p(\boldsymbol{\theta} \mid \mathbf{x})$, can be evaluated using Bayes' theorem (eq. 2.3). This probability distribution contains all current knowledge about $\boldsymbol{\theta}$.

What happens now depends on our goal. If the ultimate aim is to find the most probable value for $\boldsymbol{\theta}$, then we could produce the *maximum a posteriori* (MAP) estimate,

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \big(p(\boldsymbol{\theta} \mid \mathbf{x})\big) \tag{2.11}$$

If the priors are uniform, then this is exactly equivalent to the conventional *maximum likelihood* estimate.

This estimate does not, however, convey any information regarding the error margins or any multimodality. A better strategy is to use the whole of the posterior distribution in any further calculations.

2.1.5 Marginalisation

Most models have multiple parameters, only some of which may be of interest for solving the problem at hand. Taking the Gaussian example of §2.1.3.2, with independent priors on μ_x and σ_x^2, we can apply Bayes' theorem to obtain a joint posterior distribution:

$$p(\mu_x, \sigma_x^2 \mid \mathbf{x}) \propto p(\mathbf{x} \mid \mu_x, \sigma_x^2)\, p(\mu_x)\, p(\sigma_x^2) \tag{2.12}$$

If we are only interested in estimating μ_x, then σ_x^2 is considered to be a *nuisance parameter*, and can be *marginalised*:

$$p(\mu_x \mid \mathbf{x}) = \int p(\mu_x, \sigma_x^2 \mid \mathbf{x}) \, d\sigma_x^2 \tag{2.13}$$

$$\propto p(\mu_x) \int p(\mathbf{x} \mid \mu_x, \sigma_x^2) \, p(\sigma_x^2) \, d\sigma_x^2 \tag{2.14}$$

Bayesian inference about μ_x uses only this marginal probability distribution. If this integral cannot be performed analytically, numerical methods can be used (see Chapter 3).

2.1.6   Model selection

2.1.6.1   Motivation

Up to this point, we have assumed that we know the correct form of model for the problem. In practical modelling problems, there may be many possible models, which we have to choose between. They could be models from the same family with different numbers of parameters (such as those introduced in §2.2), or completely different types of model (such as those of §2.3). We need some means of ranking the models, in order to choose the best.

Due to noise and any random inputs to the model, a model will not fit the data precisely; there will be some random *modelling error* or *residual*. In practice, none of the available models may exactly describe the physical process which generated the data, so there can also be a systematic error.

Figure 2.2 illustrates a simple model selection problem. The candidate models are of the form

$$f_k(x, a^{(k)}) = a_1 + a_2 x + \cdots + a_k x^{k-1} \tag{2.15}$$

and each has had its parameter values, $a^{(k)}$, optimised by a least-squares procedure.

The data points were generated by a model of the same form as $f_3(x)$, but have been corrupted by additive Gaussian noise:

$$y_i = b_1 + b_2 x_i + b_3 x_i^2 + e_i \tag{2.16}$$
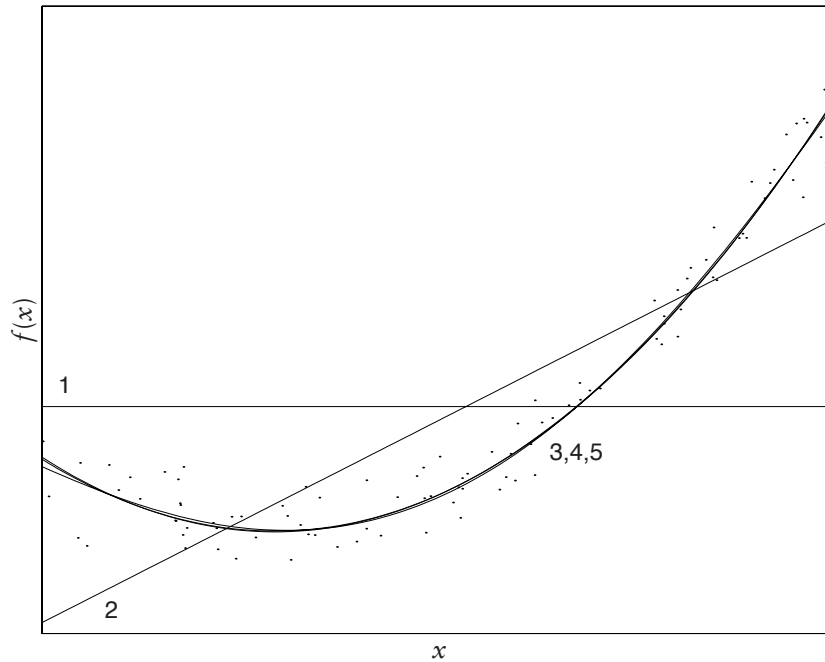
Figure 2.2. Polynomial curve-fitting problem: *100 noisy data points, gener-ated by equation* (2.16), *together with least-squares best fit curves for mod-els of order 1, 2, 3, 4 and 5.*

Table 2.1. Model selection criteria for the polynomial curve-fitting problem: *The minimum,* i.e. *best, figure in each column is marked with an asterisk.*

Model order	w.r.t. Noisy data			w.r.t. True model[a]
	RMS error	AIC	BIC	RMS error
1	15.2	38.7	41.3	14.0
2	7.41	12.7	17.9	7.76
3	1.34	6.28*	14.1*	0.372*
4	1.33	8.28	18.7	0.382
5	1.31*	10.3	23.3	0.471

[a]This measurement of model performance requires access to the true model equation (2.16), which would not be available in practice.

where

$$e_i \sim \mathrm{N}\big(e_i \mid 0, \sigma_e^2\big) \tag{2.17}$$

and $\{b_i\}$ are the correct parameter values. It is clear from the figure that the first and second order models fit very poorly, but the higher order models all seem to fit well.

The *root mean square* (RMS) error, $\hat{\sigma}_e(k)$, is one way of quantifying goodness of fit. It is defined as the square root of the error sample variance:

$$\hat{\sigma}_e^2(k) = \frac{1}{N} \sum_{i=1}^{N} \big(y_i - f(x_i, a^{(k)})\big)^2 \tag{2.18}$$

where $\{y_i\}$ are the known data points for input values $\{x_i\}$. Minimising the RMS error is equivalent to minimising the *residual sum of squares* (RSS), which is what the least-squares algorithm does.

Table 2.1 shows that the RMS error between the models' predictions and the data points decreases monotonically with increasing model order, as each additional parameter gives the model flexibility to fit the data more closely.

This does not, however, mean that the highest order model is best. The final column gives the RMS error between each candidate model's predictions and the true noise-free values at each data point. It can be seen that the third order model agrees best; the higher order ones get progressively worse.

This is called *overfitting* or *overmodelling*: the more complex models have the flexibility to model features of the limited data set which should be attributed to noise[2]. In the extreme, a model with N parameters could exactly fit all the N data points. Modelling all these features is not a good idea, as they are nothing to do with the underlying process—in another set of data from the same random process, different apparent features would appear.

Overmodelling leads to worse performance when using the model for tasks such as prediction or interpolation. The principle of *Ockham's razor*[3] is that simpler models should be favoured over more complex ones.

---

[2] or random excitation, in the case of time series models—see §2.2.
[3] "What can be done with fewer ... is done in vain with more."—William of Ockham (born *c.*1290)

2.1.6.2   Classical methods

There are various *information criteria* which are based on the RMS error but with penalties for model complexity. Several of these are special cases of the *Generalised Information Criterion* (GIC),[4]

$$\text{GIC}(k, \alpha_N) = N \ln \hat{\sigma}_e^2(k) + \alpha_N k \tag{2.19}$$

The first component decreases monotonically with increasing model order k, whereas the second increases linearly, at a rate controlled by α_N, to penalise more complex models. The relative effect of the penalty decreases with increasing data length, N. This is intuitively reasonable, as random features tend to cancel out in large data sets.

For Gaussian errors, equation (2.19) can be rewritten as

$$\text{GIC}(k, \alpha_N) = -2 \ln(\text{maximised likelihood for model } k) + \alpha_N k + c_N \tag{2.20}$$

where c_N is a constant dependent only on N.

Akaike's Information Criterion (AIC) [2] sets $\alpha_N = 2$. It is based on the information theoretic concept of entropy, has been extensively studied (see *e.g.* [99] for a brief bibliography) and is very widely used. It has, however, been shown to overestimate model orders (see *e.g.* [112]), and to be *inconsistent*, in that the probability of selecting the correct model does not approach unity as $N \to \infty$ (see *e.g.* [173]).

The *Bayesian Information Criterion* (BIC) [169] sets $\alpha_N = \ln N$. It tends to select smaller models than the AIC, and is consistent. It is an approximation to the Bayes factor (§2.1.6.3) with a particular choice of prior [112, §4.1.3].

The *Minimum Description Length* (MDL) approach to model selection [14] is motivated by coding theory. Essentially, it considers the number of bits required to describe the model, its parameter values and the modelling errors to a given precision using an optimal code. It gives rise to a criterion, of similar form to equation (2.20), which is asymptotically equivalent to the BIC but arguably better for small data sets [110].

---

[4]There are many ways to define the GIC; this one is a modification of those presented by Haughton [99] and Broersen & Wensink [25].

There are a number of other criteria and performance indices; see Gustafsson & Hjalmarsson [94], Haber & Unbehauen [95, §6.1], Koneva [116] and Dickie & Nandi [50] for reviews and summaries.

Cross validation is an alternative approach. The data set is partitioned into training and test data; each candidate model is fitted to the training data and used to predict the test data. The models are judged on the quality of their predictions as measured by the squared error. A variety of partitioning strategies are used [178, 210], some of which result in behaviour equivalent to that of the AIC and other selection criteria.

It seems reasonable to use prediction performance in model selection, given that the chosen model is likely to be used for prediction [59]. Another testing-by-prediction approach predicts one sample ahead at each point in the data set, with all past data available, to yield a criterion asymptotically equivalent to the MDL [52].

2.1.6.3 Bayesian approach

The Bayesian approach to model selection is through hypothesis testing. We want to choose the model with highest posterior probability. If we have two competing models, $\mathcal{M}_1$ and $\mathcal{M}_2$, then we can assign (perhaps equal) priors $\Pr(\mathcal{M}_1)$ and $\Pr(\mathcal{M}_2) = 1 - \Pr(\mathcal{M}_1)$. From equation (2.3), we have [112]:

$$\underbrace{\frac{\Pr(\mathcal{M}_1 \mid \mathbf{x})}{\Pr(\mathcal{M}_2 \mid \mathbf{x})}}_{\text{Posterior odds}} = \underbrace{\frac{\Pr(\mathbf{x} \mid \mathcal{M}_1)}{\Pr(\mathbf{x} \mid \mathcal{M}_2)}}_{\substack{\text{Bayes factor} \\ (B_{12})}} \underbrace{\frac{\Pr(\mathcal{M}_1)}{\Pr(\mathcal{M}_2)}}_{\text{Prior odds}} \tag{2.21}$$

where the marginal likelihoods $\Pr(\mathbf{x} \mid \mathcal{M}_i)$ are calculated by marginalising the parameter values by integrating over the parameter space (§2.1.5):

$$\Pr(\mathbf{x} \mid \mathcal{M}_i) = \int \Pr(\mathbf{x} \mid \boldsymbol{\theta}_i, \mathcal{M}_i) \, p(\boldsymbol{\theta}_i \mid \mathcal{M}_i) \, d\boldsymbol{\theta}_i \tag{2.22}$$

This differs from the previous methods, which used only one set of values of the parameters for each model—the maximum likelihood estimates. Hence the Bayes factors will depend on the priors $p(\boldsymbol{\theta} \mid \mathcal{M}_i)$.

The value of the factor B_{ij} indicates the level of evidence in favour of model j as opposed to i—the higher the value, the stronger the evidence,

independent of the prior odds. Kass & Raftery [112, §3.2] reproduce a table
of suggested thresholds for "substantial", "strong" and "decisive" evidence.

Whereas there was an explicit complexity penalty in each of the infor-
mation criteria (§2.1.6.2), the Ockham effect here is more subtle: unless
we are using a highly informative prior, as the dimension of θ increases, a
smaller proportion of the prior's probability mass falls within the region of
parameter space in which the likelihood is significant. Hence the value of
the integral of equation (2.22) falls, and so models with more parameters
are penalised.

The choice of parameter priors for Bayesian model comparison is a chal-
lenge: if the priors are too diffuse, the Ockham effect is exaggerated and
the simplest model is always chosen. This effect—that priors chosen to
be non-informative for the parameter values can strongly influence model
choice—is known as *Lindley's Paradox* [126, 175]. In the extreme case, if
the priors are improper then unknown constants appear on the top and bot-
tom of the Bayes factor; unless we have the same constants in both models'
priors, we cannot calculate the ratio.

Kass & Raftery [112, §5] describe some methods which have been devel-
oped to avoid these problems, including the use of data-dependent proper
priors, chosen to be non-informative in the region of the likelihood. A
cleaner solution, embodied in *intrinsic* [16] and *fractional* [149] Bayes fac-
tors, is to use a small part of the data as a training sample. These techniques
allow the use of an improper prior, as the unknown constants cancel. A
similar approach is used by Bishop & Djurić [22].

2.1.7 Model mixing

We saw in §2.1.4 that Bayesian parameter estimation provides a posterior
p.d.f. which contains much more information about the parameter values
and the level of uncertainty than a single estimated MAP value. This poste-
rior density can be used to marginalise unwanted parameters.

Similarly, in Bayesian model selection, the posterior probabilities of the
models under consideration form a posterior distribution, $p(\mathcal{M}_i \mid \mathbf{x})$, con-
veying more information than a simple choice of the one most probable
model. It can similarly be used to marginalise the choice of model. For

example, if our aim is to predict data **y** given data **x**, then

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=\text{all models}} p(\mathbf{y} \mid \mathbf{x}, \mathcal{M}_i)\, p(\mathcal{M}_i \mid \mathbf{x}) \qquad (2.23)$$

uses all models under consideration. This is called *model mixing* or *Bayesian model averaging*. It can give much better predictions than any single model [156].

2.2   Linear time series models

There are two basic situations in modelling:

Input/output modelling in which we have access to both the input to and output from the system, and seek to describe the function mapping from present and past (for a causal system) values of the input to the output.

Time series modelling in which all we can see is the output. In this case we want to describe the output in terms of an input/output model acting on a random, i.i.d. *excitation* process.

2.2.1   Assumptions

We will restrict ourselves to causal systems. When dealing with recording media, however, non-causal effects, such as pre-echoes caused by print-through [122], are quite possible.

Digital systems use a discrete-time (sampled) representation of the data, so it is natural to use discrete-time models, which are much simpler to implement than their continuous-time equivalents (see *e.g.* [109]). The samples are also discrete-valued, but, until Chapter 7, we will assume that the quantisation is sufficiently fine that it can be neglected.

We also initially assume that signals are zero-mean and stationary. The design of the analogue stages of most audio systems ensures that signals are usually zero-mean. Non-stationarity is addressed in §2.2.3.3 and Chapter 6.

For mathematical simplicity, the excitation is assumed to be i.i.d.

Gaussian:

$$e_t \sim p_e(e_t \mid \sigma_e^2) = \mathrm{N}(e_t \mid 0, \sigma_e^2) \tag{2.24}$$

The Gaussian distribution is appealing, as it is well understood and has many useful properties, such as being closed under convolution, which make it easy to manipulate. Some useful identities are set out in Appendix A. It also occurs frequently in nature, due to the *central limit theorem* [118, Ch. 24].

2.2.2   ARMA model

If $\{e_t\}$ is a zero-mean i.i.d. excitation process, then an *Autoregressive Moving Average* (ARMA) model [23] for a time series $\{x_t\}$ can be expressed as

$$x_t = \underbrace{\sum_{i=1}^{k} a_i x_{t-i}}_{\text{AR terms}} + \underbrace{\sum_{i=1}^{l} b_i e_{t-i}}_{\text{MA terms}} + e_t \tag{2.25}$$

where $\{x_t\}$ is the signal, $\{e_t\}$ is the excitation process, and $\{a_i; i = 1 \ldots k\}$ and $\{b_i; i = 1 \ldots l\}$ are the parameters.

This is a very general linear time series model. It incorporates both Autoregressive (AR) terms, which depend on previous values of the output, and Moving Average (MA) terms, which depend on previous values of the excitation. The above model is said to have AR terms of order k and MA terms of order l. Pure AR or MA models can be obtained as special cases of this.

AR models are also known as *all-pole* models, as the feedback terms form a filter whose z-transform has poles but no zeros. They are equivalent to *infinite impulse response* (IIR) filters. Conversely, the MA model can be thought of as *all-zero*, and is equivalent to a *finite impulse response* (FIR) filter. It is possible to fit an AR model to an MA process (or vice-versa), but to do so exactly requires a model of infinite order.

There is a problem with the ARMA model: the likelihood function is very complicated and difficult to evaluate (see *e.g.* [132]), and hence the parameters must be estimated either by search algorithms or by solving sets of nonlinear equations. There have been some recent advances by factorising the model into cascaded AR and MA processes [79, 80], but this requires

the intermediate signal (the output from the AR system, which is the input to the MA system) to be treated as an additional unknown, which requires a large amount of computation.

2.2.3   AR model

Because the response of the all-pole filter resembles the resonance of the vocal tract (see *e.g.* [48]), AR models have been used extensively in speech processing, for example in the linear predictive coding (LPC) scheme used in digital mobile telephones [170], lossless audio coding for DVD-Audio [44] and in speech enhancement [90, 124, 199]. They have been used in audio restoration, for example in the removal of clicks and crackles [78, 82, 83, 87, 89, 146–148, 158, 197, 198], and in model-based noise reduction [84–86, 88].

One possible drawback with the AR model is that the sum of two independent AR processes cannot itself be precisely represented as an AR process.[5] Hence, if a single instrument can be modelled as an AR process, it may not be reasonable to use an AR model for the whole orchestra. In practice, however, there are many useful analytic results associated with AR models—in contrast to ARMA models, parameters can be estimated in closed form—so they have been used widely even where there may not be a good physical justification.

For a purely AR model, equation (2.25) simplifies to

$$x_t = \sum_{i=1}^{k} a_i x_{t-i} + e_t \qquad\qquad (2.26)$$

The output is the excitation plus a weighted sum of past outputs at different lags. The number of parameters, k is often referred to as the *model order*.

For a signal of length n_x, we can rewrite equations (2.26) & (2.24) in matrix-vector form as

$$\mathbf{e} = \mathbf{Ax} = \mathbf{x}_1 - \mathbf{Xa} \qquad\qquad (2.27)$$

---

[5]It is an ARMA process, as is the sum of two ARMA processes [23, §A4.3].

and

$$\mathbf{e} \sim p_\mathbf{e}(\mathbf{e} \mid \sigma_e^2) = \mathbf{N}\big(\mathbf{e} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{2.28}$$

where $\mathbf{I}_{n_e}$ is the n_e-dimensional identity matrix, $\mathbf{0}$ is a column vector of zeros, $n_e = n_x - k$, and

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \quad \mathbf{x}_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \quad \mathbf{x}_1 = \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ x_{n_x} \end{bmatrix} \quad \mathbf{e} = \begin{bmatrix} e_{k+1} \\ e_2 \\ \vdots \\ e_{n_x} \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} \tag{2.29}$$

and the matrices $\mathbf{X}$ and $\mathbf{A}$ are rearrangements of $\mathbf{x}$ and $\mathbf{a}$, respectively [23]:

$$\mathbf{X} = \begin{bmatrix} x_k & x_{k-1} & \cdots & x_2 & x_1 \\ x_{k+1} & x_k & \cdots & x_3 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n_x-2} & x_{n_x-3} & \cdots & x_{n_x-k} & x_{n_x-k-1} \\ x_{n_x-1} & x_{n_x-2} & \cdots & x_{n_x-k+1} & x_{n_x-k} \end{bmatrix} \tag{2.30}$$

$$\mathbf{A} = \begin{bmatrix} -a_k & \cdots & -a_1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -a_k & \cdots & -a_1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & -a_k & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & -a_k & \cdots & -a_1 & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & -a_k & \cdots & -a_1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -a_k & \cdots & -a_1 & 1 \end{bmatrix} \tag{2.31}$$

2.2.3.1   Likelihood

We can use this notation to derive the conditional likelihood for $\mathbf{x}_1$:

$$p(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{a}, \sigma_e^2) = p_\mathbf{e}(\mathbf{e}) \tag{2.32}$$

$$= (2\pi\sigma_e^2)^{-\frac{n_e}{2}} \exp\big(-\tfrac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\big) \tag{2.33}$$

$$= (2\pi\sigma_e^2)^{-\frac{n_e}{2}} \exp\big(-\tfrac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a})\big) \tag{2.34}$$

Note that this is conditional on the initial values of the signal, $\mathbf{x}_0$. It is possible to derive $p(\mathbf{x}_0 \mid \mathbf{a}, \sigma_e^2)$, and hence to remove this conditioning (see *e.g.* [23]), but it causes problems in parameter estimation, as the expression becomes nonlinear in the parameters. For $n_x \gg k$, which is usually the case in work with audio, it is common practice to make the approximation

$$p(\mathbf{x} \mid \mathbf{a}, \sigma_e^2) \approx p(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{a}, \sigma_e^2) \tag{2.35}$$

The Gaussian in $\mathbf{Xa}$ of equation (2.34) can be transformed to a Gaussian in $\mathbf{a}$ by using the results of §A.2:

$$p(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{a}, \sigma_e^2) \propto \mathbf{N}\left(\mathbf{Xa} \mid \mathbf{x}_1, \sigma_e^2 \mathbf{I}_{n_e}\right) \tag{2.36}$$

$$\propto \mathbf{N}\left(\mathbf{a} \mid (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}_1, \sigma_e^2 (\mathbf{X}^T\mathbf{X})^{-1}\right) \tag{2.37}$$

which will be useful in parameter estimation. Clearly, the maximum likelihood estimate is

$$\hat{\mathbf{a}}_{\text{MAP}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}_1 \tag{2.38}$$

This result is the same as the least squares estimate, and can also be derived by differentiating the likelihood.

Similarly, the alternative form of the likelihood (eq. 2.27) can be transformed to be proportional to a Gaussian in $\mathbf{x}$:

$$p(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{a}, \sigma_e^2) \propto \mathbf{N}\left(\mathbf{Ax} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\right) \tag{2.39}$$

$$\propto \mathbf{N}\left(\mathbf{x} \mid \mathbf{0}, \sigma_e^2 (\mathbf{A}^T\mathbf{A})^{-1}\right) \tag{2.40}$$

which will be helpful when predicting or interpolating $\mathbf{x}$.

2.2.3.2 *Stability*

The AR model can be viewed as an IIR filter, acting on the excitation process, with the following transfer function [181, Ch. 8]:

$$H(z) = \frac{1}{A(z)} \tag{2.41}$$

where

$$
\begin{aligned}
A(z) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_k z^{-k} \\
&= z^{-k} \big((z - p_1)(z - p_2) \cdots (z - p_k) \big)
\end{aligned}
\tag{2.42}
$$

where $\{p_i\}$ are the poles of the transfer function, which, for real $\{a_i\}$, will be real or in complex conjugate pairs.

The filter, and hence the model, will be bounded-input bounded-output *stable*[6] if all the poles lie within the unit circle on the z-plane.

Although it is possible to enforce stability (see *e.g.* [137]), it is common practice simply to assume it (see *e.g.* [139]).

2.2.3.3 Time-invariance

We have, so far, assumed that the signal is *time-invariant*, *i.e.* that it has the same parameter values throughout. *Stationarity* is a stronger assumption—that all statistical characteristics are unchanging. Audio signals are usually neither time-invariant nor stationary. One way to model such changing signals is by a *time-varying model*, in which the parameters of the signal model are themselves modelled as an unobserved random process whose changes are either smooth (see *e.g.* [153, 205]) or abrupt (see *e.g.* [184, §3.3.1.7] or [40]).

Audio signals are, however, *short-term stationary*, *i.e.* short blocks can usually be considered stationary. In practice, blocks of duration less than about 25 ms seem safe [87, 88]. The audio signal can therefore be split up into short blocks, and each treated separately, except for a requirement for continuity at the block boundaries, which can be enforced by overlapping the blocks and fixing the k initial samples in each block to equal the last k samples in the previous block (see *e.g.* [88]). It may be reasonable to assume that the parameters vary slowly, in which case the posterior distribution for the parameters can be used to form the prior for the following block [79]. In certain estimation problems, some parameters are expected to remain constant across all the blocks; an approach to this is developed in Chapter 6.

---

[6]There is a confusing difference in terminology between the engineering and statistical communities: engineers refer to this as stability, whereas statisticians call it stationarity. To avoid confusion with the concept introduced in §2.2.3.3, we will use the engineering convention. In fact, for AR models, instability implies non-stationarity, but the reverse is not true.

2.3   Nonlinear time series models

2.3.1   Motivation

Linear time series models have been the subject of a huge amount of research in the 70 years since the introduction of the AR model [209]. They have an almost complete theoretical framework, and have been used successfully in countless practical applications.

However, because they model the time series as the output of a linear system, there is a variety of phenomena, including limit-cycle behaviour and time-irreversibility [184], which are observed in practice but which are not represented by linear models.

2.3.2   Volterra modelling

The Volterra series [166] is a very general means of describing a continuous-time output $\{y(t)\}$ in terms of an input $\{x(t)\}$. The Volterra series expansion for a causal, time-invariant system can be expressed as

$$y(t) = \mathbf{H}_1[x(t)] + \mathbf{H}_2[x(t)] + \mathbf{H}_3[x(t)] + \cdots + \mathbf{H}_n[x(t)] \qquad (2.43)$$

in which the n-th degree Volterra operator $\mathbf{H}_n[\cdot]$ is defined by the convolution

$$\mathbf{H}_n[x(t)] = \int_0^\infty \cdots \int_0^\infty h_n(\tau_1, \ldots, \tau_n) x(t - \tau_1) \cdots x(t - \tau_n)\, d\tau_1 \cdots d\tau_n \quad (2.44)$$

and the Volterra kernels $h_n(\cdot)$ have unspecified form, but $h_n(\tau_1, \ldots, \tau_n) = 0$ for any $\tau_i < 0, \quad i = 1, 2, \ldots, n.$

In discrete time, equation (2.44) becomes [55]

$$\mathbf{H}_n[x_t] = \sum_{j_1=0}^\infty \cdots \sum_{j_n=0}^\infty h_n(j_1, \ldots, j_n) x_{t-j_1} \cdots x_{t-j_n} \qquad (2.45)$$

This is a generalisation from linear systems theory: for a linear system, $y(t) = \mathbf{H}_1[x(t)]$, the first degree kernel $h_1(t)$ is the impulse response, which completely describes the system. For higher-degree systems, $h_n(t_1, \ldots, t_n)$ can be thought of as an n-dimensional impulse response.

Schetzen [166, §§3.4, 5.3, 5.4] develops an input/output technique for identifying the highest degree kernel $h_n(\cdot)$ by measuring the response of the system to sequences of n impulses with varying spacing. He then recursively identifies $h_{n-1}(\cdot)$ in the same way.

This representation leads to an explosive growth of the number of co-efficients with the degree of the model or the number of lags considered. Morrison [143, §§2.2, 2.5] considers several methods to combat this, including triangularising the matrix representation of the kernels, excluding cross-terms and neglecting small-valued coefficients. For audio applications, Reed & Hawksford [160, 161] describe simplifications which can be used where the linear response is known to dominate over nonlinear components.

There are problems with the use of Volterra series:

- The Schetzen [166] algorithm mentioned above requires knowledge of the highest degree Volterra operator present in the system. Furthermore, its complexity increases dramatically with degree.

  For systems excited by (observable) white Gaussian noise, these problems are overcome by the use of Wiener G-functionals [166, Ch. 12], which are mutually orthogonal and hence can be determined independently.

- The response of the system in equation (2.43) to the input $x(t) = a\,u(t)$ is:

$$y(t) = \sum_{n=1}^{\infty} a^n \, \mathbf{H}_n[u(t)] \tag{2.46}$$

  It can be seen that it behaves as a power series. It therefore suffers similar convergence problems to the Taylor series: expansions may only converge for a small range of inputs (see *e.g.* [166, pp. 200–202]).

The kernels of a Volterra model can be treated either as an infinite number of coefficients, a vast number of which may need to be stored to give a good approximation, or as functions.

Discrete Volterra models are widely used in the control literature (see *e.g.* [21]), classification problems (see *e.g.* [157]) and artificial neural networks [31, 128]. Present applications in audio include input/output

modelling of audio systems (see *e.g.* [160]) and nonlinear filtering to pre-compensate for known loudspeaker nonlinearities [115, 168].

2.3.3   Parametric nonlinear models

An alternative methodology for nonlinear modelling is to use models with similar structures to those of the parametric linear time series models introduced in §2.2.

There is a plethora of such models, but no universally recognised method to categorise them. For example, Tong [184, Ch. 3], Tjøstheim [183], and Chen & Billings [38] take radically different approaches. They can all, however, be treated as generalisations or specialisations of the nonlinear ARMA model.

2.3.3.1   Nonlinear ARMA model

The ARMA model of equation (2.25) can be generalised to give the Nonlinear ARMA (NARMA) model. If we restrict ourselves to additive noise, this takes the form [37]

$$x_t = f(x_{t-1}, \ldots, x_{t-k}, e_{t-1}, \ldots, e_{t-l}) + e_t \qquad (2.47)$$

where $f(\cdot)$, rather than being a simple weighted sum, as was the case in linear modelling, is now some arbitrary nonlinear function, of which there are uncountably many from which to choose. Nonlinear AR (NAR) and MA (NMA) models are obvious simplifications. We now look at some popular choices for $f(\cdot)$.

2.3.3.2   Bilinear model

Defining

$$f(\cdot) = a_0 + \sum_{i=1}^{A} a_i x_{t-i} + \sum_{i=1}^{B} b_i e_{t-i} + \sum_{i=1}^{C} \sum_{j=1}^{D} c_i d_j x_{t-i} e_{t-j} \qquad (2.48)$$

gives the bilinear model, which contains all the terms of the ARMA model plus bilinear product terms, so called because they depend linearly on both the input and the output; if both are varying then the effect is nonlinear. The above model is denoted $BL(A, B, C, D)$ [32].

2.3.3.3 *Polynomial model*

The ARMA model of equation (2.25) is a polynomial of degree one. The bilinear model incorporates some second degree cross-product terms. It seems natural to use polynomials of higher degrees as a systematic means to approximate unknown nonlinearities [37, 133].

Clearly, if the summations in the Volterra model of equation (2.45) are truncated to finite length and finite maximum degree, it becomes a polynomial nonlinear MA model.

But there is a problem: for large values at the input, it is *explosive*—*i.e.* the output starts to diverge. Chen & Billings [37] argue that, although this may prevent the use of polynomial models for simulation, they can certainly be used for *n*-step-ahead prediction, provided n is not too large. Tong [184, pp. 103–107] suggests using *censoring*, either by bounding the output of the polynomial to the range $\pm Q$, or by using a threshold model (see §2.3.3.7) to replace the polynomial function with a linear one, $l(\mathbf{q})$, outside a certain range of the inputs:

$$\tilde{f}(\mathbf{q}) = \begin{cases} f(\mathbf{q}) & \text{for } \|\mathbf{q}\| \leq R \\ l(\mathbf{q}) & \text{for } \|\mathbf{q}\| > R \end{cases} \tag{2.49}$$

where $\mathbf{q}$ contains all relevant variables and $\|\cdot\|$ is some suitable norm.

2.3.3.4 *LITP models*

A *linear in the parameters* (LITP) (or *pseudo-linear regression*) model is one in which $f(\cdot)$ can be expressed in the form

$$f(\cdot) = \sum_{i=1}^{k} \theta_i z_i(\cdot) \tag{2.50}$$

where the θ_i are parameters and the *regressors* or *basis functions* $z_i(\cdot)$ are arbitrary (but fixed, for a given model) functions of past inputs and outputs. It can be seen that bilinear, polynomial and Volterra models all fall within this class. Other LITP models (not discussed here) include radial basis functions and single layer neural nets. The advantage of using a model of this form is that many techniques developed for linear models can still be used (see *e.g.* [151]).

2.3.3.5 Additive models

Clearly, even the double summation in equation (2.48) can lead to very large numbers of parameters, and introducing higher powers makes this worse—the so-called *curse of dimensionality*. Additive models [36] assume no interaction between different lags, *i.e.*

$$f(\cdot) = f_1(x_{t-1}) + f_2(x_{t-2}) + \cdots + f_A(x_{t-A}) \tag{2.51}$$

Chen & Tsay [36] argue that, although additivity is a strong assumption, the *nonlinear additive AR* (NAAR) model is sufficiently general for many applications. Chen *et al.* [34] provide methods to check for additivity.

2.3.3.6 Exponential AR models

Another possibility is to introduce arbitrary nonlinear components to the function, perhaps supported by physical arguments. One such is the exponential ARMA (EXPARMA) model [37]:

$$f(\cdot) = \sum_{i=1}^{A} \left(a_i + b_i \exp(-x_{t-i}^2) \right) + \sum_{i=1}^{C} \left(c_i + d_i \exp(-x_{t-i}^2) \right) e_{t-i} \tag{2.52}$$

Although nonlinear near $x_t = 0$, this reverts smoothly to linearity as $x_t \to \pm\infty$.

2.3.3.7 Threshold models

In a threshold model [184, §3.3], different functions $f(\cdot)$ are used depending on the value of the output at some fixed lag d. This introduces nonlinearities even when the functions themselves are linear. It can be written as

$$f(\cdot) = \begin{cases} g^{(1)}(\cdot) & \text{if } r_0 < x_{t-d} < r_1 \\ g^{(2)}(\cdot) & \text{if } r_1 < x_{t-d} < r_2 \\ \vdots & \vdots \\ g^{(q)}(\cdot) & \text{if } r_{q-1} < x_{t-d} < r_q \end{cases} \tag{2.53}$$

where the thresholds, r_i, satisfy

$$-\infty = r_0 < r_1 < \cdots < r_{q-1} < r_q = \infty \tag{2.54}$$

and the $g^{(i)}$ can be defined as for any of the previously mentioned linear or nonlinear models.

It can be seen that the EXPARMA model discussed earlier is a form of threshold model with a smooth transition over the threshold. Another smooth threshold model is the logistic smooth threshold AR model (LSTAR) [180].

2.3.3.8 *Functional coefficient models*

Chen & Tsay [35] introduce a different generalisation: they keep the same form of $f(\cdot)$, say AR, for all inputs, but make each parameter a function of past inputs. Clearly, the threshold model can be described in this way, by formulating an $f(\cdot)$ which incorporates all the $g^{(i)}(\cdot)$ and indicators to switch between them if necessary.

2.3.3.9 *Doubly stochastic models*

We can go further than this by treating the parameters as the outputs of stochastic processes. In the *Random Coefficient AR* (RCA) model, each coefficient is drawn from an i.i.d. process. The parameters of this process could be derived from the current state of an (unobserved) Markov chain [184, §3.3.1.7]. The number of states in the chain and the transition probabilities would then need to be inferred. Clearly, there is a risk of vastly over-parameterising the model such that there may never be enough data to allow estimation.

2.3.3.10 *State-space models*

In a *state-space representation* (see *e.g.* [30, 171, 204]), a time series model is formulated as a combination of a *state equation* and an *observation* or *measurement equation*. The state equation takes an excitation input and gives an output (that we cannot observe) which is an input to the measurement equation, which takes a separate excitation or noise input and generates the output (that we *can* observe!).

There is a variety of powerful algorithms that can be applied to linear Gaussian state-space models, including the Kalman filter and Kalman smoother. See Wu [208, Ch. 7] for an excellent overview.

In the linear state-space literature, the state equation usually has AR
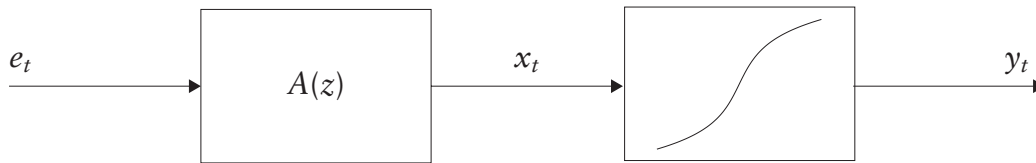
Figure 2.3. Wiener cascade model: *A linear system driving a memoryless nonlinearity.*

form, and the measurement equation MA. Carlin, Polson & Stoffer [29] generalise these to nonlinear functions—NAR and NMA respectively—to provide a flexible framework for forecasting and filtering using nonlinear state-space models with possibly non-Gaussian excitation.

2.3.4   Cascade models

Rather than using large, general nonlinear models, an alternative approach is to cascade smaller models together, connecting the output of one to the input of the next. This can correspond to the stages of the system itself; for example Mercer [141] suggests using three stages in modelling distorted audio signals:

1. A linear system (with random excitation) to represent the original signal;

2. A memoryless nonlinearity (MNL) to represent the distortion introduced by the recording medium;

3. A further linear system to represent the equalisation circuitry through which the signal will have passed *after* being distorted.

This is known as a Wiener-Hammerstein model [95]: the Hammerstein model [91] consists of just an MNL followed by a linear system; the Wiener model is the other way round: a linear system whose output passes through an MNL (fig. 2.3). Both the Wiener and Hammerstein models can be linear in the parameters if the component models themselves are.

Block-oriented models are a generalisation of cascade models to allow arbitrary connections, including feedback and feedforward, between subsystems. They are widely used in the control literature.

2.4   Discussion

This chapter has introduced the concepts behind Bayesian modelling which underlie all that follows, and then discussed linear and nonlinear time series models. The AR model will be used as a signal model in Chapters 4, 6 & 7 and, since it is simple, but quite general, the Volterra polynomial NAR model will be used in Chapters 5 & 6. State-space methods will be used in §4.7, and different forms of cascade model in Chapters 6 & 7.

Markov chain
Monte Carlo methods

3

3.1 Motivation

As shown in the previous chapter, it is often necessary to integrate functions over a high-dimensional probability distribution in order to perform Bayesian inference. For example, to find the expected value of $f(\boldsymbol{\theta})$, where $f(\cdot)$ is some arbitrary function whose expectation exists:

$$E\left[f(\boldsymbol{\theta})\right] = \int \cdots \int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{3.1}$$

Unless $f(\cdot)$ and $p(\cdot)$ take convenient functional forms, or can be closely approximated, it is not usually possible to perform this integral analytically: numerical methods must be used.

The most straightforward approach to numerical integration is to discretise the parameter space, $\boldsymbol{\theta}$, at some finite resolution, so that the multidimensional integral can be approximated by a multiple summation. To illustrate this in a two dimensional, square parameter space:

$$E\left[f(\boldsymbol{\theta})\right] \approx \sum_{i=1}^{M} \sum_{j=1}^{M} f\left(\boldsymbol{\theta}(i,j)\right) p\left(\boldsymbol{\theta}(i,j)\right) \Delta^2 \tag{3.2}$$

where

$$\Delta = \tfrac{1}{M}(\theta^{\max} - \theta^{\min}) \tag{3.3}$$

and

$$\boldsymbol{\theta}(i,j) = \begin{bmatrix} \theta^{\min} + i\Delta \\ \theta^{\min} + j\Delta \end{bmatrix} \tag{3.4}$$

The problem with this approach is that for good accuracy a small Δ, and hence large M, must be used, but the computation involved is $O(M^D)$, where D is the dimension of $\boldsymbol{\theta}$.

Due to the computation required, such deterministic numerical integration becomes impractical at moderately high dimensions. There are many variations on this basic technique (see *e.g.* [148]), but most have similar limitations. We therefore pursue an alternative approach.

3.2   Monte Carlo integration

Returning to the expectation of equation (3.1), if we have a set of samples $\{\boldsymbol{\theta}_n; n = 1 \ldots N\}$ drawn from the distribution $p(\boldsymbol{\theta})$, then we can estimate the value of the integral by a Monte Carlo estimate,

$$E\left[f(\boldsymbol{\theta})\right] \approx \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{\theta}_n) \tag{3.5}$$

This is performing stochastic numerical integration. By the laws of large numbers, this estimate can be improved to any required level of accuracy by increasing the sample size N.

This approach works for any $f(\boldsymbol{\theta})$. However, unless $p(\boldsymbol{\theta})$ takes a convenient form, it may be difficult to draw samples from it. Two solutions to this are importance sampling and Markov chains.

3.2.1   Importance sampling

If we cannot obtain samples from $p(\boldsymbol{\theta})$ directly, the conceptually simplest solution is to draw samples from some *similar* distribution, say $\pi(\boldsymbol{\theta})$, then compensate for the difference in the distributions by *importance reweighting* [97, §5.4]:

$$E\left[f(\boldsymbol{\theta})\right] \approx \frac{\sum_{n=1}^{N} w_n f(\boldsymbol{\theta}_n)}{\sum_{n=1}^{N} w_n} \tag{3.6}$$

where

$$w_n = \frac{p(\boldsymbol{\theta}_n)}{\pi(\boldsymbol{\theta}_n)} \tag{3.7}$$

This method is typically used to reduce the variance of Monte Carlo estimates by focusing the sampling distribution on the region of interest [76].

A drawback with this approach is that, for complex $p(\boldsymbol{\theta})$, it can be difficult to find an easily-sampled distribution which is sufficiently similar to $p(\boldsymbol{\theta})$. If $\pi(\boldsymbol{\theta}) \ll p(\boldsymbol{\theta})$ for some values of $\boldsymbol{\theta}$, then that part of the distribution may be represented by only a few samples, with correspondingly very high weights, resulting in poor accuracy. Rejection sampling (§3.4.2) is a related approach, with similar limitations.

3.2.2   Markov chains

The alternative approach, which we will pursue, is to draw non-i.i.d. samples from the correct distribution, $p(\boldsymbol{\theta})$. These can be produced using a *Markov chain*.

A Markov chain[1] is a discrete-time random process which can be modelled as a state machine in which the probability of moving to a new state depends *only* on the current state. The transition probabilities for a discrete-valued process $\{x(n); n = 0, \dots\}$ can hence be expressed as a single *transition matrix*,

$$\mathbf{T} = \begin{bmatrix} \Pr(S_1 \to S_1) & \dots & \Pr(S_k \to S_1) \\ \vdots & \ddots & \vdots \\ \Pr(S_1 \to S_k) & \dots & \Pr(S_k \to S_k) \end{bmatrix} \tag{3.8}$$

whose columns sum to 1, and where k is the number of states and $Pr(S_a \to S_b)$ is the probability of moving to state S_b in the next step if the current state is S_a.

The state probabilities at step n can then be calculated as

$$\mathbf{s}(n) = \begin{bmatrix} \Pr\big(x(n) = S_1\big) \\ \vdots \\ \Pr\big(x(n) = S_k\big) \end{bmatrix} = \mathbf{T}^n \mathbf{s}(0) \tag{3.9}$$

where $\mathbf{s}(0)$ is the initial distribution.

All Markov chains have a *stationary distribution* such that

$$\mathbf{T}\mathbf{s}^{\text{stat}} = \mathbf{s}^{\text{stat}} \tag{3.10}$$

---

[1] Named after A. A. Markov (1856-1922), who used them to model character and word sequences in text.

where $\mathbf{s}^{\text{stat}}$ is an eigenvector of $\mathbf{T}$ with eigenvalue 1. It can be shown [144] that, iff $\mathbf{T}$ has *exactly* one eigenvalue equal to 1, and no complex eigenvalues with magnitudes greater than or equal to 1, then the Markov chain has a unique *limiting distribution*, such that

$$\lim_{n \to \infty} \mathbf{s}(n) = \mathbf{s}^{\text{lim}} \tag{3.11}$$

independent of the initial distribution. Similar results hold for continuous-valued processes [144, 182].

3.3 *Markov chain Monte Carlo methods*

The Markov chain Monte Carlo (MCMC) approach to evaluating the expectation of equation (3.1) is to construct a Markov chain with $\boldsymbol{\theta}$ as its state variable and $p(\boldsymbol{\theta})$ (the *target distribution*) as its limiting distribution. This chain is then simulated, from an arbitrary initial state, to provide samples $\{\boldsymbol{\theta}_n; n = 1 \ldots N\}$. If enough samples are generated, then the distribution of the later samples will tend towards $p(\boldsymbol{\theta})$.[2] These are then used in equation (3.5) to make a Monte Carlo estimate of the required expectation.

3.3.1 Metropolis-Hastings algorithm

To study the effect of different molecular potential fields on the properties of fluids, Metropolis, Rosenbluth, Rosenbluth, Teller & Teller [142] model the fluid as N particles in a finite space[3]. To estimate quantities of interest, it is necessary to integrate over all possible configurations of the particles, according to their probabilities, which are functions of the potential energy.

To obtain samples from the configuration space of the particles, for use in making a Monte Carlo estimate of the multidimensional integral, the system is simulated. For each particle in turn, a move is proposed, in which the particle is translated through a random displacement sampled from a simple bounded uniform density. This move is then either accepted or rejected,

---

[2] Samples at the beginning of the chain will be dependent on the initial state, so they are often discarded as *burn-in*. An alternative strategy is to choose an initial state which is expected to be typical of those produced by the chain [73].
[3] Strictly speaking, they use a periodic space in order to eliminate boundary effects.

depending on its effect on the potential energy of the system, in order to produce samples whose potential energy follows a Boltzmann distribution.

Hastings [98] generalises this to produce a method for constructing Markov chains with any desired limiting distribution $p(\boldsymbol{\theta})$, with arbitrary, including asymmetric, proposal distributions.

The chain starts in an arbitrary initial state $\boldsymbol{\theta}(0)$. At each step, a new state $\boldsymbol{\theta}'(n+1)$ is proposed by sampling from some convenient *proposal distribution* $q\big(\boldsymbol{\theta}'(n+1) \mid \boldsymbol{\theta}(n)\big)$. An *acceptance probability*,

$$\alpha\big(\boldsymbol{\theta}(n) \to \boldsymbol{\theta}'(n+1)\big) = \min\left(1, \underbrace{\frac{p\big(\boldsymbol{\theta}'(n+1)\big)}{p\big(\boldsymbol{\theta}(n)\big)}}_{\substack{\text{Target} \\ \text{probability ratio}}} \underbrace{\frac{q\big(\boldsymbol{\theta}(n) \mid \boldsymbol{\theta}'(n+1)\big)}{q\big(\boldsymbol{\theta}'(n+1) \mid \boldsymbol{\theta}(n)\big)}}_{\substack{\text{Transition proposal} \\ \text{probability ratio}}}\right) \qquad (3.12)$$

is calculated. Then, with probability $\alpha(\boldsymbol{\theta}(n) \to \boldsymbol{\theta}'(n+1))$, the proposed state is accepted, and the chain moves: $\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}'(n+1)$. Otherwise, the chain remains in the same state: $\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n)$.

This form of acceptance probability ensures that the *reversibility condition* [98] is satisfied:

$$p(\boldsymbol{\theta}_A)\, q(\boldsymbol{\theta}_B \mid \boldsymbol{\theta}_A)\, \alpha(\boldsymbol{\theta}_A \to \boldsymbol{\theta}_B) = p(\boldsymbol{\theta}_B)\, q(\boldsymbol{\theta}_A \mid \boldsymbol{\theta}_B)\, \alpha(\boldsymbol{\theta}_B \to \boldsymbol{\theta}_A) \qquad (3.13)$$

i.e. the probability of being in state A and moving to state B is equal to that of being in state B and moving to state A.[4] Together with irreducibility and aperodicity, this is sufficient to ensure that the stationary distribution is $p(\boldsymbol{\theta})$ and the chain will converge to that, independent of the initial state [144].

A useful feature of the Metropolis-Hastings algorithm is that the target distribution need only be known up to a constant of proportionality, as the acceptance probability only uses the ratio $\frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta})}$.

In equation (3.12), all components of $\boldsymbol{\theta}$ are updated at once (a *global* update [144]). This need not be the case: Metropolis *et al.* [142] sample pairs of coordinates (associated with a single particle) in rotation (fixed scan, local update), and Hastings [98] allows for arbitrary *blocking* of components with either fixed or random *scanning*.

---

[4]For continuous distributions, *detailed balance*, an equivalent integral form of this condition, is used [144].

Where multiple blocks of components are used, the acceptance probability becomes:[5]

$$\alpha \left(\begin{bmatrix} \boldsymbol{\theta}_{[i]} \\ \boldsymbol{\theta}_{[-i]} \end{bmatrix} \rightarrow \begin{bmatrix} \boldsymbol{\theta}'_{[i]} \\ \boldsymbol{\theta}_{[-i]} \end{bmatrix} \right) = \min \left(1, \frac{p\left(\boldsymbol{\theta}'_{[i]} \mid \boldsymbol{\theta}_{[-i]}\right)}{p\left(\boldsymbol{\theta}_{[i]} \mid \boldsymbol{\theta}'_{[-i]}\right)} \frac{q\left(\boldsymbol{\theta}_{[i]} \mid \boldsymbol{\theta}'\right)}{q\left(\boldsymbol{\theta}'_{[i]} \mid \boldsymbol{\theta}\right)} \right) \qquad (3.14)$$

where the notation $\boldsymbol{\theta}_{[i]}$ and $\boldsymbol{\theta}_{[-i]}$ denotes partitioning into those elements currently being sampled and the remainder, which remain fixed, and $p\left(\boldsymbol{\theta}_{[i]} \mid \boldsymbol{\theta}_{[-i]}\right)$ is the *full conditional distribution* for the component $\boldsymbol{\theta}_{[i]}$.

All that is required for convergence is that, as the number of iterations tends to infinity, so each component will tend to have been sampled infinitely often [182]. The relative merits of different blocking and scanning patterns are discussed in §3.5.3.

The proposal distribution, $q\left(\boldsymbol{\theta}'_{[i]} \mid \boldsymbol{\theta}\right)$, can be any distribution from which it is convenient to sample. Algorithms can be separated into three broad classes according to the components on which the proposal distribution is dependent.

3.3.1.1 Metropolis sampler

The original scheme of Metropolis *et al.* [142] uses a proposal distribution of the form

$$q\left(\boldsymbol{\theta}'_{[i]} \mid \boldsymbol{\theta}\right) = g\left(\left|\boldsymbol{\theta}'_{[i]} - \boldsymbol{\theta}_{[i]}\right|\right) \qquad (3.15)$$

where $g(\cdot)$ is an arbitrary function. This is known as a *random-walk sampler*.

In the *Metropolis sampler*, which is a generalisation of this, the proposal distribution can take any form which is *symmetric*, *i.e.*

$$q\left(\boldsymbol{\theta}' \mid \boldsymbol{\theta}\right) = q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}'\right) \qquad (3.16)$$

so that the terms in equation (3.14) involving $q(\cdot)$ cancel, leaving just the ratio of target (conditional) probabilities.

The variance of the proposal distribution is important: if it is too narrow,

---

[5]For clarity, the (n) and $(n+1)$ arguments are neglected from here on, except where they are necessary to avoid ambiguity.

the chain will not move quickly through the parameter space; if it is too broad, most proposed moves will have low acceptance probabilities. Both cases lead to slow convergence.

3.3.1.2   Independence sampler

In the *independence sampler* [98, 182], the proposal distribution is fixed, independent of $\boldsymbol{\theta}$. Where there is detailed knowledge about the likely parameter values, this can be used to construct a proposal distribution which will concentrate sampling in that area. For example, in sampling for the frequencies of component sinusoids, the Fourier transform of the signal could be used as a proposal distribution (see [7] and §7.3). If the likely parameter distribution can be very closely approximated, it might be possible to use importance sampling (§3.2.1), which has the advantage of producing i.i.d. samples.

3.3.1.3   Gibbs sampler

The *Gibbs sampler* [61] is a special case of the Metropolis-Hastings algorithm, in which the proposal distribution for each component is that component's full conditional distribution:

$$q\left(\boldsymbol{\theta}'_{[i]} \mid \boldsymbol{\theta}\right) = p\left(\boldsymbol{\theta}_{[i]} \mid \boldsymbol{\theta}_{[-i]}\right) \tag{3.17}$$

This makes the $q(\cdot)$ terms in the acceptance probability (eq. 3.14) cancel the $p(\cdot)$ terms, so that the acceptance probability becomes 1, *i.e.* all moves are accepted. This greatly simplifies implementation.

Used in image restoration, the original implementation of the Gibbs sampler samples a single pixel's value at a time, in a fixed scan. Geman & Geman [65] suggest that, since the full conditional distribution for a pixel under their Markov random field image model is dependent only on the values of neighbouring pixels, a parallel implementation should be possible, giving great speed advantages.

A Gibbs sampler in which several components are sampled jointly, *i.e.* the partition $(\cdot)_{[i]}$ contains several terms, is sometimes referred to as a *multi-move* Gibbs sampler (see §3.5.3.2).

3.3.2   Reversible-jump MCMC

Reversible-jump MCMC [92] is a generalisation of the Metropolis-Hastings algorithm to cases where the dimensionality of the parameter space is itself an unknown parameter. Jump diffusion [93] is a similar approach. This problem arises in model selection: candidate models may have differing numbers of parameters.

In addition to standard Metropolis-Hastings or Gibbs sampler moves, new moves are introduced which jump between parameter spaces of differing dimension. The form of the acceptance probability is such that detailed balance (see §3.3.1) is satisfied within each class of move.

The reversible-jump methodology can be illustrated using the Bayesian model selection problem of §2.1.6.3, where there are multiple models of different orders, which are being used to fit the data $\mathbf{x}$. The k parameters associated with the model of order k are represented by the vector $\boldsymbol{\theta}^{(k)}$.

Standard Metropolis-Hastings or Gibbs sampler moves can be used to sample $\boldsymbol{\theta}^{(k)}$ while staying within the same model.

A new type of move can be proposed to go from a model of order k to one of order k', with probability $J(k' \mid k)$. This is called the *move probability*, and incorporates the probability of choosing to propose this type of move. If $k' > k$ then this is called a *birth move*; if $k > k'$ then it is a *death move*.

The derivation of the acceptance probability [92] introduces a *dimension matching* requirement. If the proposed new parameter value, $\boldsymbol{\theta}^{(k')}$, is calculated as a deterministic function of the existing parameter value, $\boldsymbol{\theta}^{(k)}$, and an additional random vector,

$$\mathbf{u}^{(k \to k')} \sim q(\mathbf{u}^{(k \to k')}) \tag{3.18}$$

of dimension $m^{(k \to k')}$, and $k + m^{(k \to k')} = k' + m^{(k' \to k)}$, then the requirement is met [92].

This leads to an acceptance probability of the form

$$
\alpha\left((k, \boldsymbol{\theta}^{(k)}) \to (k', \boldsymbol{\theta}^{(k')})\right)
$$
$$
= \min\left(1, \underbrace{\frac{p\left(k', \boldsymbol{\theta}^{(k')} \mid \mathbf{x}\right)}{p\left(k, \boldsymbol{\theta}^{(k)} \mid \mathbf{x}\right)}}_{\substack{\text{Target probability} \\ \text{ratio}}} \underbrace{\frac{J\left(k \mid k'\right)}{J\left(k' \mid k\right)}}_{\substack{\text{Move proposal} \\ \text{probability ratio}}} \underbrace{\frac{q\left(\mathbf{u}^{(k' \to k)}\right)}{q\left(\mathbf{u}^{(k \to k')}\right)}}_{\substack{\text{Random vector} \\ \text{proposal ratio}}} \underbrace{\left| \frac{\partial\left(\boldsymbol{\theta}^{(k')}, \mathbf{u}^{(k' \to k)}\right)}{\partial\left(\boldsymbol{\theta}^{(k)}, \mathbf{u}^{(k \to k')}\right)} \right|}_{\text{Jacobian}}\right) \tag{3.19}
$$

3.3.3 Simulated annealing

In metallurgy, *annealing* is a means to allow a material to reach its lowest energy state, which consists of large crystals. The material is heated to a very high temperature, so that molecules have sufficient kinetic energy to overcome any energy barriers, then gradually cooled, following an *annealing schedule*.

In annealed MCMC [144, §6.1], the temperature, T, becomes a parameter of the stationary distribution, *e.g.*

$$\pi_T(\boldsymbol{\theta}) = p(\boldsymbol{\theta})^{\frac{1}{T}} \tag{3.20}$$

Hence the acceptance probability function and (optionally) the proposal distribution also depend on T. At $T = 1$, the acceptance probability is the same as in standard MCMC, so samples are produced from the target posterior distribution. At high values of T, the stationary distribution is flattened, and so the acceptance probability tends to 1 and the chain can move freely through the state space, regardless of the target distribution. At $T = 0$, only moves to lower energy (higher probability) states are accepted, so the chain behaves like a deterministic gradient descent algorithm, converging on the nearest minimum.

During cooling, the chain can hop between modes by jumping over the energy barrier. Following an annealing schedule until T is almost 0 will exaggerate the target distribution and hence converge on a mode. Under certain conditions, it has been shown that annealing schedules of the form $T \propto \frac{1}{\log n}$ are guaranteed to converge to the global minimum [65]. Unfortunately, such logarithmic schedules are much too slow to be of practical use. Popular optimisation techniques use much faster schedules, which cannot be similarly justified [144].

Stopping cooling at $T = 1$ allows samples to be generated from the $p(\boldsymbol{\theta})$, as in standard MCMC. Multiple runs should be more likely to explore all parts of the parameter space than in standard MCMC.

In *Metropolis-coupled MCMC* [76, §6.4.2], one chain is run at $T = 1$, and others at a series of higher temperatures. Metropolis-Hastings moves are included which can swap states between the chains, enabling the base chain to benefit from the good mixing properties of the higher temperature chains, whilst still having the correct stationary distribution. A development

of this, known as *simulated tempering* [74], saves computation by running only one of these chains at a time, but introduces moves which allow a different temperature to be chosen. Samples produced by chains other than the base chain are discarded.

Another approach is simply to run the chain at some temperature $T > 1$, then use importance sampling (§3.2.1) to reweight the samples from $\pi_T(\boldsymbol{\theta})$ to evaluate expectations over $p(\boldsymbol{\theta})$. For T reasonably close to 1, this gives the advantage of better mixing whilst still retaining accuracy.

3.4   *Sampling difficulties*

Given a source of random or pseudo-random numbers from a uniform distribution, there is a wide range of distributions from which it is straightforward to produce samples, including Gaussian and Gamma distributions (see *e.g.* [105, 155]), and transformations of these such as inverse Gamma. It is often, however, desirable to use different distributions as proposal distributions in MCMC systems; Gibbs samplers, for example, require samples from the full conditional distribution. Methods for producing samples from arbitrary distributions include inverse c.d.f. methods and rejection sampling.

3.4.1   Inverse c.d.f. methods

If the cumulative distribution function (c.d.f.), $F(\theta)$, of a desired univariate proposal distribution $p(\theta)$ is known and can be inverted, then uniform samples,

$$x \sim \mathrm{U}(x \mid 0, 1) \qquad (3.21)$$

can be transformed to i.i.d. samples from $p(\theta)$ as follows:

$$\theta = F^{-1}(x) \qquad (3.22)$$

If $F(\theta)$ is not available, it can be approximated by evaluating $p(\theta)$ at a number of points (a *grid*) $\theta = \phi_1, \phi_2, \ldots \phi_n$, then interpolating between these points. When used to sample from full conditionals for the Gibbs sampler, Tanner [179] calls this method *Griddy Gibbs*. With the use of non-uniform and adaptive grids and judiciously chosen interpolation methods, good results can be obtained.

3.4.2 Rejection sampling

Rejection sampling, first proposed by von Neumann [200], is an alternative approach which can readily be generalised to produce samples from multivariate distributions.

An *envelope distribution*, $q(\boldsymbol{\theta})$, is chosen, which is similar in shape to the target distribution, $p(\boldsymbol{\theta})$, but from which it is convenient to sample. A *scaling factor*, s, is found which is the minimum value which satisfies

$$s\,q(\boldsymbol{\theta}) \geq p(\boldsymbol{\theta}) \qquad \forall\,\boldsymbol{\theta} \tag{3.23}$$

A value is proposed by drawing

$$\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}) \tag{3.24}$$

then an acceptance probability is calculated:

$$\alpha_{\mathrm{rs}}(\boldsymbol{\theta}') = \frac{p(\boldsymbol{\theta}')}{s\,q(\boldsymbol{\theta}')} \tag{3.25}$$

With probability $\alpha_{\mathrm{rs}}(\boldsymbol{\theta}')$, the proposed value is accepted, and output as a sample from $p(\boldsymbol{\theta})$. Otherwise, it is rejected and new values must be repeatedly proposed until one is accepted.

Although apparently similar to the independence sampler (§3.3.1.2) rejection sampling has the advantage that it produces i.i.d. samples, whereas rejected moves in the independence sampler lead to correlation. It has, however, a lower acceptance rate.[6]

---

[6]The acceptance probability for the independence sampler with the same proposal distribution is

$$\alpha_{\mathrm{indep}}(\boldsymbol{\theta} \to \boldsymbol{\theta}') = \min\left(1, \frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta})}{q(\boldsymbol{\theta}')}\right) \tag{3.26}$$

From equation (3.23),

$$s \geq \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \qquad \forall\,\boldsymbol{\theta} \tag{3.27}$$

using this with equation (3.25) and cancelling the terms involving $\boldsymbol{\theta}'$ gives

$$\alpha_{\mathrm{rs}}(\boldsymbol{\theta}) \leq \alpha_{\mathrm{indep}}(\boldsymbol{\theta} \to \boldsymbol{\theta}') \qquad \forall\,\boldsymbol{\theta},\boldsymbol{\theta}' \tag{3.28}$$

We have not seen this result presented elsewhere.

It can be very difficult to construct an envelope function which is good enough to give acceptably low rejection rates. If no standard distributions, or mixtures thereof, are close enough, a piecewise approximation can be attempted. In *adaptive rejection sampling* (ARS), a piecewise exponential envelope function is improved, by dividing one of the pieces, each time a sample is drawn [77, §5.3.3]. *Adaptive rejection Metropolis sampling* (ARMS) [77, §5.3.5], extends ARS to non-log-concave distributions by allowing the envelope function to fall short of the target distribution in some places, then correcting for this using a Metropolis-Hastings step.

The use of rejection sampling, and other methods, to sample from truncated Gaussian distributions is discussed in §7.2.3.2.

3.5   *Convergence*

The various MCMC methods which have been described construct Markov chains which can be shown to converge to the correct stationary distribution, independent of the starting point, as the number of iterations tends to infinity. Running the chain for an infinite number of iterations is not an option, so we need to decide how many iterations should be discarded as burn-in, and how many subsequent iterations should be used to obtain an acceptable level of accuracy.

3.5.1   Theory

For certain target distributions, and chains complying with certain conditions, geometric convergence can be guaranteed and there are lower bounds on the rate of convergence [6, 150]. Unfortunately, these bounds tend to be extremely conservative: in practical MCMC, much faster convergence is often observed. Recent results are, however, becoming more realistic [163].

3.5.2   Diagnostics

In the absence of useful theoretical guidance, many tools have been developed to help assess whether a chain has converged (see *e.g.* [140] for a review).

The first problem is to decide at what point the chain has started producing samples (approximately) from the target distribution. The most common approach is to use some measure of stationarity [98]. The assumption is that, if the chain's output appears to be stationary, then the chain must be in its unique stationary distribution, which is the target distribution.

The problem with this approach is that there may be narrow peaks, perhaps of significant probability mass, in the target distribution, which the sampler has not yet encountered. Mengersen *et al.* [140] describe this as the "you've only seen where you've been" principle. Gelman & Rubin [63, 64] suggest that this risk can be lessened by running multiple chains, with different initial values drawn from some approximation to the target distribution. When all the chains seem to have converged to the same distribution, they argue that it is likely to be the right one.

The drawback of this multiple chain approach is that *each* of the m chains takes n iterations to converge, requiring $m \times n$ iterations to be computed before samples can start to be drawn for use in the estimation. Geyer [72] argues that this is both wasteful and unlikely to detect failure to converge for some (possibly degenerate) distributions. He argues that expending that amount of computation on a single chain is more likely to lead to convergence.

In deciding how many iterations are needed, after convergence, for use in the Monte Carlo estimate, the correlation between samples leads to the standard results, which assume i.i.d. samples, providing an underestimate [182]. Simulation from multiple chains here has an advantage: the different chains are independent, which should reduce the required number of samples.

MCMC convergence is a highly active research area, but it is not the focus of this thesis. In the work that follows, convergence is monitored simply by plotting the evolution of either the parameter values or the overall performance in reconstruction tasks. In making inference, as many samples as possible are used. We are generally only interested in samplers which converge quickly, as, given the huge amount of data involved in audio processing, it is not practical to compute the huge numbers of iterations commonly used by the statistical community.

3.5.3 Faster convergence

Despite the difficulty in predicting convergence rates and detecting convergence, there are some techniques which have been shown to speed convergence.

3.5.3.1 *Scanning patterns*

Samplers with random scan patterns are more amenable to theoretical analysis, as their output is both reversible and acyclic [26, 131].

It has been observed that, for some classes of sampler and target distribution, random scanning leads to faster convergence than fixed scanning [176]. Roberts & Sahu [163], however, show that this is not a general rule: *e.g.* for Gaussian target distributions, the best choice of scanning pattern depends on the target covariance matrix.

3.5.3.2 *Correlated components*

When highly correlated components are only sampled separately, convergence tends to be slow [130, 131]. This can usually be mitigated by reparameterisation [60], or by blocking—*i.e.* sampling the highly correlated parameters jointly—where possible [131]. Roberts & Sahu [163] demonstrate that the effects of blocking are dependent on the problem; they show examples where blocking slightly slows convergence.

3.6 *Summary*

This chapter has highlighted the need for numerical methods in Bayesian analysis and the limitations of direct numerical integration, then introduced several variants of the MCMC approach, on which the following chapters rely. It has also considered some of the implementation details, such as procedures for drawing random samples from arbitrary distributions (which are investigated further in Chapter 7), means for diagnosing misconvergence and techniques which tend to speed convergence.

Model order uncertainty

<div align="right">

4

</div>

4.1 Motivation

In time series modelling problems, the model order is usually unknown. Section 2.1.6.2 described commonly-used criteria for judging the best model order. Choosing a model order is not often the ultimate aim of a modelling exercise. More commonly, there is some other task to perform in the presence of model uncertainty, such as prediction, interpolation or noise reduction.

In this chapter, concentrating on the AR model, we examine means to allow for model order uncertainty in an MCMC framework, introducing the possibility of *model mixing*, in which estimates are based on a combination of different models according to their posterior probabilities. This is explored, for the case of audio noise reduction, in §4.7.

4.2 Model selection using MCMC

As discussed in §2.1.6.3, given data $\mathbf{x}$ and candidate models $\{\mathcal{M}^{(k)}; k = 1 \ldots k_{\max}\}$, each with parameters $\boldsymbol{\theta}^{(k)}$, Bayesian model selection is performed on the basis of the posterior model probabilities, $p(\mathcal{M}^{(k)} \mid \mathbf{x})$.

These can be evaluated using MCMC by treating the model index, k, as an unknown parameter, and so updating it at each iteration. The obvious next step is to set up a Markov chain to generate samples from $p(k, \boldsymbol{\theta}^{(k)} \mid \mathbf{x})$ then use these samples to make a Monte Carlo estimate of $p(k)$.

There is a problem with this approach: if the models under comparison have differing numbers of parameters, then $p(k, \boldsymbol{\theta}^{(k)})$ will be a probability measure over spaces of varying dimension, which cannot be compared.

4.2.1   Carlin & Chib

This problem can be overcome by ensuring that the dimension remains constant, whichever model is being considered. Carlin & Chib [28] do this by sampling from a *composite model space* consisting of all of the parameters of all the models under consideration, *i.e.* they produce samples from $p(k, \boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(1)}, \ldots \boldsymbol{\theta}^{(k_{\max})})$.

They use the Gibbs sampler to sample from the joint distribution. For simplicity, we assume that the components of each parameter vector are sampled jointly; smaller blocks may be used in practice. Each model's parameter vector is sampled independently, from its full conditional distribution:

$$\boldsymbol{\theta}^{(j)} \sim \begin{cases} p(\boldsymbol{\theta}^{(k)} \mid \mathbf{x}, \mathcal{M}^{(k)}) & \text{if } j = k \\ \rho(\boldsymbol{\theta}^{(j)} \mid \mathcal{M}^{(k)}) & \text{if } j \neq k \end{cases} \tag{4.1}$$

where $\rho(\cdot)$ is called the *pseudo-prior* [28], which can be any proper distribution, as $\boldsymbol{\theta}^{(j)}$ has no effect on the model when $k \neq j$.

The model index is sampled, in another Gibbs move, from

$$k \sim p(\mathcal{M}^{(k)} \mid \mathbf{x}, \boldsymbol{\theta}^{(1)}, \ldots \boldsymbol{\theta}^{(k_{\max})}) \tag{4.2}$$

which can be constructed from

$$\Pr(\mathcal{M}^{(k)} \mid \mathbf{x}, \boldsymbol{\theta}^{(1)}, \ldots \boldsymbol{\theta}^{(k_{\max})})$$
$$= \frac{p(\mathbf{x} \mid \boldsymbol{\theta}^{(k)}, \mathcal{M}^{(k)}) \left\{ \prod_{i=1}^{k_{\max}} p(\boldsymbol{\theta}^{(i)} \mid \mathcal{M}^{(k)}) \right\} p(\mathcal{M}^{(k)})}{\sum_{j=1}^{k_{\max}} \left(p(\mathbf{x} \mid \boldsymbol{\theta}^{(j)}, \mathcal{M}^{(j)}) \left\{ \prod_{i=1}^{k_{\max}} p(\boldsymbol{\theta}^{(i)} \mid \mathcal{M}^{(j)}) \right\} p(\mathcal{M}^{(j)}) \right)} \tag{4.3}$$

To speed convergence, Carlin & Chib suggest choosing pseudo-priors which approximate the full conditional distribution for the parameters, *i.e.* $\rho(\boldsymbol{\theta}^{(j)} \mid \mathcal{M}^{(k)}) \approx p(\boldsymbol{\theta}^{(j)} \mid \mathbf{x}, \mathcal{M}^{(j)})$.

As an implementation note, they suggest that, if one of the models under consideration has very high probability, leading to slow convergence, its prior probability can be reduced to encourage better mixing; as the Bayes factor is calculated from the ratio of posterior to prior probabilities, it will not be affected.

This is not an efficient method, as all parameters of all models must be sampled at each iteration. Godsill [81] considers the issues of composite models and pseudo-priors in detail.

This approach has the advantage that it can be used to compare models of completely different structure. Where the models are sufficiently similar that some of the parameters, ϕ, appear with the same interpretation in all the models, computation could be reduced by sampling only one instance of ϕ, the value of which is shared by all the models. In *nested* models, all the parameters of a lower order model can be incorporated, with the same interpretation, in higher order models. In the model of order k, $\mathcal{M}^{(k)}$, the parameter vector $\boldsymbol{\theta}^{(k)}$ has k elements, although there may be additional shared parameters in ϕ, such as noise variances which are common to all models.

We now consider a model selection method which can make use of the relationship between the parameters of models of different orders.

4.2.2 Reversible-jump

Rather than using a Gibbs sampler, the composite model space of §4.2.1 can be sampled using Metropolis-Hastings moves (§3.3.1). Godsill [81] shows that, if new values $(k', \boldsymbol{\theta}'^{(1)}, \ldots \boldsymbol{\theta}'^{(k_{\max})})$ are proposed by drawing from independent proposal distributions

$$k' \sim J(k' \mid k) \tag{4.4}$$

$$\boldsymbol{\theta}'^{(j)} \sim \begin{cases} q(\boldsymbol{\theta}'^{(k')} \mid \mathbf{x}, \mathcal{M}^{(k')}, \boldsymbol{\theta}^{(k)}) & \text{if } j = k' \\ \rho(\boldsymbol{\theta}'^{(j)}) & \text{if } j \neq k' \end{cases} \tag{4.5}$$

where $J(\cdot)$ and $q(\cdot)$ are arbitrary proposal distributions and $\rho(\cdot)$ is a pseudo-prior, then all terms involving parameters of models other than $\mathcal{M}^{(k)}$ and $\mathcal{M}^{(k')}$ cancel from the acceptance probability, leaving

$$\alpha\big((k, \boldsymbol{\theta}^{(1)}, \ldots \boldsymbol{\theta}^{(k_{\max})}) \to (k', \boldsymbol{\theta}'^{(1)}, \ldots \boldsymbol{\theta}'^{(k_{\max})})\big)$$

$$= \alpha\big((k, \boldsymbol{\theta}^{(k)}) \to (k', \boldsymbol{\theta}'^{(k')})\big) \tag{4.6}$$

$$= \min\left(1, \frac{p(\mathcal{M}^{(k')}, \boldsymbol{\theta}'^{(k')} \mid \mathbf{x})}{p(\mathcal{M}^{(k)}, \boldsymbol{\theta}^{(k)} \mid \mathbf{x})} \frac{J(k \mid k')}{J(k' \mid k)} \frac{q(\boldsymbol{\theta}^{(k)} \mid \boldsymbol{\theta}'^{(k')})}{q(\boldsymbol{\theta}'^{(k')} \mid \boldsymbol{\theta}^{(k)})}\right) \tag{4.7}$$

Hence the parameters of models $\{\mathcal{M}^{(j)}; j \notin \{k, k'\}\}$ are not used, and so need not be sampled at all.

Equation (4.7) is equivalent to the reversible-jump acceptance probability (eq. 3.19), except that there is no Jacobian term, as the new parameter values are proposed directly rather than as a function of the old parameter values and a separate random vector [92]. Dellaportas, Forster & Ntzoufras [47] also note the relationship between composite models and reversible-jump.

4.2.3   Application to AR model order

Much previous work on AR model order uncertainty using MCMC has reparameterised the model in terms of partial autocorrelation coefficients, $\mathbf{r}$. This has the advantage that the region in $\mathbf{r}$-space which corresponds to stable AR models is a simple cuboid, from which samples can easily be drawn. There is a one-to-one mapping between $\mathbf{r}^{(k)}$ and $\mathbf{a}^{(k)}$.

Barnett, Kohn & Sheather [13] include all coefficients up to a specified maximum order, then perform subset selection by associating a binary indicator variable with each coefficient, in a similar manner to that described in Chapter 5.

Barbieri & O'Hagan [10] and Andrieu, Doucet & Duvaut [8] work within a reversible-jump framework. They use only two types of model order move: $(\mathcal{M}^{(k)}, \mathbf{r}^{(k)}) \to (\mathcal{M}^{(k+1)}, \mathbf{r}^{(k+1)})$ and $(\mathcal{M}^{(k)}, \mathbf{r}^{(k)}) \to (\mathcal{M}^{(k-1)}, \mathbf{r}^{(k-1)})$. The AR model is treated as a nested model under the new parameterisation, so those elements of $\mathbf{r}$ which are common to both the current and proposed models remain unchanged.

Huerta & West [104] instead reparameterise the model in terms of the positions of individual real poles and pairs of complex conjugate poles. In many applications, this allows straightforward decomposition of the signal into components with different physical significance.

4.3   Modelling framework

An alternative approach is to use the natural parameterisation of the model. The advantage of this over the methods of §4.2.3 is that it allows greater

analytic simplification of the resulting algorithms. At present, stability is not enforced; for the applications of §4.7 and Chapters 6 & 7, this has not proved to be a problem.

This section describes the model and Bayesian framework; §4.4 will consider the design of the reversible-jump sampler.

4.3.1 Autoregressive model

From §2.2.3, a signal $\mathbf{x}$ can be modelled as an AR process with white Gaussian excitation as follows:

$$x_t = e_t + \sum_{i=1}^{k} a_i^{(k)} x_{t-i} \tag{4.8}$$

where

$$e_t \overset{\text{i.i.d.}}{\sim} \mathrm{N}\big(e_t \mid 0, \sigma_e^2\big) \tag{4.9}$$

is the excitation sequence and $\mathbf{a}^{(k)}$ is the AR parameter vector for a kth order model. This can be rewritten in matrix-vector form as

$$\mathbf{e} = \mathbf{A}\mathbf{x} = \mathbf{x}_1 - \mathbf{X}^{(k)}\mathbf{a}^{(k)} \tag{4.10}$$

where $\mathbf{x}_0$ and $\mathbf{x}_1$ are formed by partitioning $\mathbf{x}$ into, respectively, the first k values and the remainder[1], and $\mathbf{A}$ and $\mathbf{X}^{(k)}$ take appropriate forms.

Since the excitation sequence is Gaussian, the (approximate) likelihood takes the form [23, §A7.4]

$$p(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2) \approx p(\mathbf{x}_1 \mid \mathbf{x}_0, k, \mathbf{a}^{(k)}, \sigma_e^2) \tag{4.11}$$

$$= \mathrm{N}\big(\mathbf{e} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{4.12}$$

$$= (2\pi\sigma_e^2)^{-\frac{n_e}{2}} \exp\big(-\tfrac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\big) \tag{4.13}$$

where n_e is the length of $\mathbf{e}$ and $\mathbf{x}_1$.

---

[1]Because it is necessary in later experiments to ensure continuity between adjacent blocks of the signal, a different partitioning is used for experiments: $\mathbf{x}_1$ contains the current block of the signal, of fixed length, and $\mathbf{x}_0$ contains k initial values, taken from the end of the previous block. For clarity, the more conventional partitioning is used in descriptions and derivations.

4.3.2   Prior distributions

As discussed in §2.1.3, prior distributions represent knowledge of or belief about parameter values before examining the data. k, $\mathbf{a}^{(k)}$, σ_a^2 and σ_e^2 are assumed to be *a priori* independent. For the model order, we choose a simple bounded uniform distribution:

$$p(k) = \begin{cases} \frac{1}{k_{\max}+1} & \text{if } k \in \{0, 1, \ldots k_{\max}\} \\ 0 & \text{elsewhere} \end{cases} \tag{4.14}$$

where $k_{\max}$ is set high enough to have no effect on model selection. If more specific prior model probabilities are available, they can be used instead.

In order to perform Bayesian model selection, we must use proper priors for the model parameters (see §2.1.6.3). For the AR parameters, we use the conjugate prior, which is a multivariate Gaussian distribution:

$$p(\mathbf{a}^{(k)} \mid \sigma_a^2) = \mathbf{N}\big(\mathbf{a}^{(k)} \mid \boldsymbol{\mu}_{p\mathbf{a}^{(k)}}, \mathbf{C}_{p\mathbf{a}^{(k)}}\big) \tag{4.15}$$

In the absence of any genuine prior knowledge, we set $\boldsymbol{\mu}_{p\mathbf{a}^{(k)}} = \mathbf{0}$ and $\mathbf{C}_{p\mathbf{a}^{(k)}} = \sigma_a^2 \mathbf{I}_k$, but any multivariate Gaussian could be used without significant changes to the MCMC framework.

The excitation variance is assigned its conjugate prior,

$$p(\sigma_e^2) = \mathrm{IG}\big(\sigma_e^2 \mid \alpha_e, \beta_e\big) \tag{4.16}$$

where the inverse Gamma distribution [108] is defined for positive parameters α and β, and positive θ, as

$$\mathrm{IG}(\theta \mid \alpha, \beta) \propto \theta^{-(\alpha+1)} \exp(-\beta/\theta) \tag{4.17}$$

which tends to the uninformative Jeffreys' prior for scale parameters as $\alpha, \beta \to 0$.

The hyperparameter σ_a^2 is assigned a similar inverse Gamma prior, which is again conjugate:

$$p(\sigma_a^2) = \mathrm{IG}\big(\sigma_a^2 \mid \alpha_a, \beta_a\big) \tag{4.18}$$

4.3.3 Bayesian hierarchy

The joint posterior distribution for the model parameters can then be obtained using Bayes' theorem (eq. 2.3) as

$$p(k, \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2 \mid \mathbf{x}) \propto \underbrace{p(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2)}_{\text{Likelihood}} \underbrace{p(k) \, p_{\mathbf{a}}(\mathbf{a}^{(k)} \mid \sigma_a^2) \, p(\sigma_a^2) \, p(\sigma_e^2)}_{\text{Priors}} \quad (4.19)$$

4.4 Reversible-jump sampling strategies

It is straightforward to sample σ_e^2 and σ_a^2, which are common to all the candidate models, using Gibbs sampler moves, which will be described in §4.5.1.

We now develop three possible approaches to reversible-jump sampling of the model order, k, and corresponding parameters $\mathbf{a}^{(k)}$.

4.4.1 Straightforward approach

In his paper introducing reversible-jump sampling [92], Green describes a simple method for designing moves between models which satisfy the "dimension-matching" requirement (see §3.3.2). It can be applied to the AR model as follows:

4.4.1.1 Birth move

For a *birth* move, *i.e.* from a model of order k to one of order $k' = k + n$, a value of $\mathbf{a}^{(k')}$ should be calculated as a deterministic function of $\mathbf{a}^{(k)}$ and $\mathbf{u}^{(k \to k')}$, where $\mathbf{u}^{(k \to k')}$ is a random vector of dimension n. Since this is a nested model, the simplest way to do this is to leave the common parameters of the two models unchanged, and draw the values of the proposed new parameters from some other distribution, such as their prior.

To simplify notation, we denote these new parameters $\mathbf{a}_u^{(k')}$ rather than $\mathbf{u}^{(k \to k')}$, such that $(\cdot)_u$ denotes the parameters which are being *updated*. $\mathbf{a}_f^{(k')}$ contains the rest of the parameters, which remain *fixed* under the proposed

move, as follows:

$$\mathbf{a}_u^{(k')} \sim \overbrace{p(\mathbf{a}_u^{(k')} \mid \sigma_a^2)}^{\text{Prior}} \tag{4.20}$$

$$\mathbf{a}^{(k')} = \begin{bmatrix} \mathbf{a}_f^{(k')} \\ \mathbf{a}_u^{(k')} \end{bmatrix} = \begin{bmatrix} \mathbf{a}^{(k)} \\ \mathbf{a}_u^{(k')} \end{bmatrix} \tag{4.21}$$

4.4.1.2   Death move

The corresponding *death* move, from order k to order $k' = k - n$, is purely deterministic:

$$\mathbf{a}^{(k')} = \mathbf{a}_{\langle 1 \dots k - n \rangle}^{(k)} \tag{4.22}$$

where the notation $(\cdot)_{\langle a \dots b \rangle}$ represents a vector containing the components with indices between a and b.

4.4.1.3   Acceptance probabilities

Birth or death moves are proposed by sampling $k' \sim J(k \to k')$. For simplicity, a symmetric random-walk proposal distribution (§3.3.1.1) is used.

Their acceptance probability follows directly from equation (4.7):

$$\alpha\big((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)$$
$$= \min\left(1, \underbrace{\frac{p\big(k', \mathbf{a}^{(k')} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}{p\big(k, \mathbf{a}^{(k)} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}}_{\substack{\text{Ratio of posteriors}}} \underbrace{\frac{J(k \mid k')}{J(k' \mid k)}}_{\substack{\text{Ratio of model} \\ \text{transition proposal} \\ \text{probabilities}}} \underbrace{\frac{q\big(\mathbf{a}_u^{(k)} \mid \sigma_a^2\big)}{q\big(\mathbf{a}_u^{(k')} \mid \sigma_a^2\big)}}_{\substack{\text{Ratio of} \\ \text{random vector} \\ \text{proposal} \\ \text{probabilities}}}\right) \tag{4.23}$$

Since the transformation of equation (4.21) does not involve a change in scale, there is no Jacobian term. Also, as model transitions are proposed from a symmetric random-walk distribution, the terms involving $J(\cdot)$ cancel:

$$\alpha\big((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)$$
$$= \min\left(1, \frac{p\big(k', \mathbf{a}^{(k')} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}{p\big(k, \mathbf{a}^{(k)} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)} \frac{q\big(\mathbf{a}_u^{(k)} \mid \sigma_a^2\big)}{q\big(\mathbf{a}_u^{(k')} \mid \sigma_a^2\big)}\right) \tag{4.24}$$

Furthermore, the ratio of posteriors can be expressed in terms of the likelihood (eq. 4.13) and priors (eq. 4.14 & 4.15) using Bayes' theorem (eq. 2.3):

$$
\alpha\left((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\right)
$$
$$
= \min\left(1, \frac{p\left(\mathbf{x} \mid k', \mathbf{a}^{(k')}, \sigma_e^2\right)}{p\left(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2\right)} \frac{p\left(k'\right)}{p\left(k\right)} \frac{p\left(\mathbf{a}^{(k')} \mid \sigma_a^2\right)}{p\left(\mathbf{a}^{(k)} \mid \sigma_a^2\right)} \frac{q\left(\mathbf{a}_u^{(k)} \mid \sigma_a^2\right)}{q\left(\mathbf{a}_u^{(k')} \mid \sigma_a^2\right)}\right) \quad (4.25)
$$

where unnecessary conditioning has been dropped. Note the prior independence of k and $\mathbf{a}^{(k)}, \sigma_a^2$. Since $p(k)$ is uniform over all permissible models (eq. 4.14), it cancels $p(k')$. Also, since $p(\mathbf{a}^{(k)} \mid \sigma_a^2)$ is simply i.i.d. Gaussian (eq. 4.15), the priors for the common model terms cancel, leaving only (for a birth move)

$$
\alpha\left((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\right)
$$
$$
= \min\left(1, \frac{p\left(\mathbf{x} \mid k', \mathbf{a}^{(k')}, \sigma_e^2\right)}{p\left(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2\right)} p_{\mathbf{a}}\left(\mathbf{a}_u^{(k')} \mid \sigma_a^2\right) \frac{q\left(\mathbf{a}_u^{(k)} \mid \sigma_a^2\right)}{q\left(\mathbf{a}_u^{(k')} \mid \sigma_a^2\right)}\right) \quad (4.26)
$$

For the forward transition in a birth move, $\mathbf{a}_u^{(k')}$ is drawn from the prior on $\mathbf{a}_u$, *i.e.*

$$
q\left(\mathbf{a}_u^{(k')} \mid \sigma_a^2\right) = p\left(\mathbf{a}_u^{(k')} \mid \sigma_a^2\right) \quad (4.27)
$$

whereas the reverse transition is deterministic, so $\mathbf{a}_u^{(k)}$ is empty and $q\left(\mathbf{a}_u^{(k)}\right) = 1$. The prior and proposal terms thus cancel, simplifying the acceptance probability to

$$
\alpha\left((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\right) = \min\left(1, \frac{p\left(\mathbf{x} \mid k', \mathbf{a}^{(k')}, \sigma_e^2\right)}{p\left(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2\right)}\right) \quad (4.28)
$$

which applies for either a birth or death move.

4.4.2   Proposing new parameters from full conditionals

A more elaborate approach, which makes better use of analytic results available for the AR model, is to propose new model parameters from their full conditional distributions. Due to the choice of a conjugate prior, the full conditional is a multivariate Gaussian distribution, from which it is

straightforward to sample.

With this proposal distribution, the birth move becomes

$$\mathbf{a}_u^{(k')} \sim \overbrace{p(\mathbf{a}_u^{(k')} \mid \mathbf{x}, k', \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2)}^{\text{Full conditional}} \tag{4.29}$$

$$\mathbf{a}^{(k')} = \begin{bmatrix} \mathbf{a}_f^{(k')} \\ \mathbf{a}_u^{(k')} \end{bmatrix} = \begin{bmatrix} \mathbf{a}^{(k)} \\ \mathbf{a}_u^{(k')} \end{bmatrix} \tag{4.30}$$

The death move remains the same (eq. 4.22).

With this partitioning, the AR process (eq. 4.10) can be rewritten as

$$\mathbf{e} = \mathbf{x}_1 - \mathbf{X}^{(k')}\mathbf{a}^{(k')} = \mathbf{x}_1 - \mathbf{X}_f^{(k')}\mathbf{a}_f^{(k')} - \mathbf{X}_u^{(k')}\mathbf{a}_u^{(k')} \tag{4.31}$$

The Jacobian is again unity, so, from equation (4.24), the acceptance probability is

$$\alpha\big((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)$$

$$= \min\left(1, \frac{p\big(k', \mathbf{a}^{(k')} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}{p\big(k, \mathbf{a}^{(k)} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)} \frac{1}{p\big(\mathbf{a}_u^{(k')} \mid k', \mathbf{a}_f^{(k')}, \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}\right) \tag{4.32}$$

Rather than simply drawing a value of $\mathbf{a}_u^{(k')}$ from the full conditional, then evaluating this acceptance probability, we can use the "Candidate's Identity", which is a simple result from probability theory [20]:

$$\frac{p(k, \boldsymbol{\theta} \mid \mathbf{x})}{p(\boldsymbol{\theta} \mid k, \mathbf{x})} = p(k \mid \mathbf{x}) \tag{4.33}$$

to simplify equation (4.32) to

$$\alpha\big(k \to k' \mid \mathbf{x}, \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2\big) = \min\left(1, \frac{p\big(k' \mid \mathbf{x}, \mathbf{a}_f^{(k')}, \sigma_a^2, \sigma_e^2\big)}{p\big(k \mid \mathbf{x}, \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2\big)}\right) \tag{4.34}$$

where $\mathbf{a}_f^{(k')} = \mathbf{a}^{(k)}$ (eq. 4.30). The acceptance probability is hence independent of $\mathbf{a}_u^{(k')}$, so a value need not be drawn unless the move is accepted. The expression in the numerator, from which $\mathbf{a}_u^{(k')}$ has been analytically

marginalised, is derived in §B.1 as

$$p(k' \mid \mathbf{x}, \mathbf{a}_f^{(k')}, \sigma_a^2, \sigma_e^2) \tag{4.35}$$

$$= \int p(k', \mathbf{a}_u^{(k')} \mid \mathbf{x}, \mathbf{a}_f^{(k')}, \sigma_a^2, \sigma_e^2) \, d\mathbf{a}_u^{(k')} \tag{4.36}$$

$$\propto \frac{\sqrt{|\mathbf{C}_{c a_u^{(k')}}|}}{(\sqrt{2\pi}\sigma_e)^{n_e} \sqrt{|\mathbf{C}_{p a_u^{(k')}}|}} \exp\left(-\frac{1}{2\sigma_e^2}\left[\mathbf{e}_f^T \mathbf{e}_f - \frac{1}{\sigma_e^2}\mathbf{e}_f^T \mathbf{X}_u^{(k')} \mathbf{C}_{c a_u^{(k')}} \mathbf{X}_u^{(k')T} \mathbf{e}_f\right]\right) \tag{4.37}$$

where

$$\mathbf{C}_{c a_u^{(k')}} = \left(\frac{1}{\sigma_e^2}\mathbf{X}_u^{(k')T}\mathbf{X}_u^{(k')} + \mathbf{C}_{p a_u^{(k')}}^{-1}\right)^{-1} \tag{4.38}$$

$$\boldsymbol{\mu}_{c a_u^{(k')}} = \frac{1}{\sigma_e^2}\mathbf{C}_{c a_u^{(k')}}\mathbf{X}_u^{(k')T}\mathbf{e}_f \tag{4.39}$$

$$\mathbf{e}_f = \mathbf{x}_1 - \mathbf{X}_f^{(k')}\mathbf{a}_f^{(k')} \tag{4.40}$$

such that $\mathbf{e}_f$ is the excitation signal corresponding to a model containing only the terms associated with the parameters $\mathbf{a}_f^{(k')}$.

The acceptance probability (eq. 4.34) then becomes

$$\alpha\left(k \rightarrow k' \mid \mathbf{x}, \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2\right)$$

$$= \min\left(1, \sqrt{\frac{|\mathbf{C}_{c a_u^{(k')}}|}{|\mathbf{C}_{p a_u^{(k')}}|}} \exp\left(\frac{1}{2\sigma_e^4}\mathbf{e}_f^T\mathbf{X}_u^{(k')}\mathbf{C}_{c a_u^{(k')}}\mathbf{X}_u^{(k')T}\mathbf{e}_f\right)\right) \tag{4.41}$$

4.4.3 Proposing whole parameter vector

An alternative approach is to ignore the nested nature of the model and propose a complete new parameter vector, discarding the current value. This should allow better mixing, as the chain can move between any two models in a single step.

With this proposal distribution, $\mathbf{a}^{(k)}$ is not used, so birth and death moves are treated identically:

$$\mathbf{a}^{(k')} \sim p(\mathbf{a}^{(k')} \mid \mathbf{x}, k', \sigma_a^2, \sigma_e^2) \tag{4.42}$$

The acceptance probability follows from equation (4.24):

$$\alpha\big((k, \mathbf{a}^{(k)}) \to (k', \mathbf{a}^{(k')}) \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)$$
$$= \min\left(1, \frac{p\big(k', \mathbf{a}^{(k')} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}{p\big(k, \mathbf{a}^{(k)} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)} \frac{p\big(\mathbf{a}^{(k)} \mid \mathbf{x}, k, \sigma_a^2, \sigma_e^2\big)}{p\big(\mathbf{a}^{(k')} \mid \mathbf{x}, k', \sigma_a^2, \sigma_e^2\big)}\right) \quad (4.43)$$

As in §4.4.2, there is no need to draw a value of $\mathbf{a}^{(k')}$ unless the move is accepted, as it cancels from the expression for the acceptance probability, leaving

$$\alpha\big(k \to k' \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big) = \min\left(1, \frac{p\big(k' \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}{p\big(k \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)}\right) \quad (4.44)$$

which is independent of both $\mathbf{a}^{(k)}$ and $\mathbf{a}^{(k')}$. The derivation follows the same lines as that in §B.1 and §4.4.2, but with the partitioning changed such that $(\cdot)_u$ contains the whole parameter vector and $(\cdot)_f$ is empty, and results in

$$\mathbf{C}_{c\mathbf{a}^{(k')}} = \big(\tfrac{1}{\sigma_e^2}\mathbf{X}^{(k')T}\mathbf{X}^{(k')} + \mathbf{C}_{p\mathbf{a}^{(k')}}^{-1}\big)^{-1} \quad (4.45)$$

$$\boldsymbol{\mu}_{c\mathbf{a}^{(k')}} = \tfrac{1}{\sigma_e^2}\mathbf{C}_{c\mathbf{a}^{(k')}}\mathbf{X}^{(k')T}\mathbf{x}_1 \quad (4.46)$$

and

$$\alpha\big(k \to k' \mid \mathbf{x}, \sigma_a^2, \sigma_e^2\big)$$
$$= \min\left(1, \sqrt{\frac{|\mathbf{C}_{c\mathbf{a}^{(k')}}|}{|\mathbf{C}_{c\mathbf{a}^{(k)}}|} \frac{|\mathbf{C}_{p\mathbf{a}^{(k)}}|}{|\mathbf{C}_{p\mathbf{a}^{(k')}}|}} \frac{\exp\big(\tfrac{1}{2}\boldsymbol{\mu}_{c\mathbf{a}^{(k')}}^T \mathbf{C}_{c\mathbf{a}^{(k')}}^{-1} \boldsymbol{\mu}_{c\mathbf{a}^{(k')}}\big)}{\exp\big(\tfrac{1}{2}\boldsymbol{\mu}_{c\mathbf{a}^{(k)}}^T \mathbf{C}_{c\mathbf{a}^{(k)}}^{-1} \boldsymbol{\mu}_{c\mathbf{a}^{(k)}}\big)}\right) \quad (4.47)$$

Note that the same acceptance probability could be obtained by marginalising $\mathbf{a}^{(k)}$ directly before designing the model moves.

It is possible to marginalise σ_e^2 as well, by reparameterising such that $\sigma_a^2 = \nu\,\sigma_e^2$ (see *e.g.* [7]). Whilst this is mathematically convenient, it is not physically plausible—we expect σ_a^2 and σ_e^2 to be independent since, for example, scaling the signal will vary σ_e^2 but not σ_a^2.

4.5 Implementation

4.5.1 Other sampling steps

4.5.1.1 Sampling the AR parameter vector

We can sample $\mathbf{a}^{(k)}$ directly from its full conditional distribution in a Gibbs move, for which the acceptance probability is always one (§3.3.1.3):

$$\mathbf{a}^{(k)} \sim p(\mathbf{a}^{(k)} \mid \mathbf{x}, k, \sigma_a^2, \sigma_e^2) \tag{4.48}$$

$$= \mathbf{N}\big(\mathbf{a}^{(k)} \mid \boldsymbol{\mu}_{c\mathbf{a}^{(k)}}, \mathbf{C}_{c\mathbf{a}^{(k)}}\big) \tag{4.49}$$

where $\boldsymbol{\mu}_{c\mathbf{a}^{(k)}}$ and $\mathbf{C}_{c\mathbf{a}^{(k)}}$ are defined in a similar manner to equations (4.45) & (4.46). This is equivalent to a model move of §4.4.3 in which $k' = k$. This step should be performed in algorithms using the moves of §4.4.1 and §4.4.2 in order to allow the lower order AR parameters to be updated.

4.5.1.2 Sampling the noise variance

We can also sample σ_e^2 using a Gibbs move. To do this, we require the full conditional posterior distribution:

$$p(\sigma_e^2 \mid \mathbf{x}, k, \mathbf{a}^{(k)}, \sigma_a^2) \propto \overbrace{p(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_a^2, \sigma_e^2)}^{\text{Likelihood}} \overbrace{p(\sigma_e^2)}^{\text{Prior}} \tag{4.50}$$

$$\approx \mathbf{N}\big(\mathbf{e}^T\mathbf{e} \mid 0, \sigma_e^2\mathbf{I}_{n_e}\big) \, \mathrm{IG}\big(\sigma_e^2 \mid \alpha_e, \beta_e\big) \tag{4.51}$$

$$\propto (2\pi\sigma_e^2)^{-\frac{n_e}{2}} \exp\big(-\tfrac{1}{2\sigma_e^2}\mathbf{e}^T\mathbf{e}\big) \sigma_e^{-2(\alpha_e+1)} \exp\big(-\tfrac{\beta_e}{\sigma_e^2}\big) \tag{4.52}$$

$$= \sigma_e^{-(n_e+2\alpha_e+2)} \exp\big(-\tfrac{\beta_e+\frac{1}{2}\mathbf{e}^T\mathbf{e}}{\sigma_e^2}\big) \tag{4.53}$$

$$= \mathrm{IG}\big(\sigma_e^2 \mid \alpha_{se}, \beta_{se}\big) \tag{4.54}$$

where

$$\alpha_{se} = \alpha_e + \tfrac{1}{2}n_e \qquad \text{and} \qquad \beta_{se} = \beta_e + \tfrac{1}{2}\mathbf{e}^T\mathbf{e} \tag{4.55}$$

We can sample from this inverse Gamma density directly (see *e.g.* [155]).

4.5.1.3   Sampling the parameter variance

Similarly, we can use a Gibbs move to sample the hyperparameter σ_a^2:

$$p(\sigma_a^2 \mid \mathbf{x}, k, \mathbf{a}^{(k)}, \sigma_e^2) = p(\sigma_a^2 \mid \mathbf{a}^{(k)}) \tag{4.56}$$

$$\propto p(\mathbf{a}^{(k)} \mid \sigma_a^2)\, p(\sigma_a^2) \tag{4.57}$$

$$= \mathbf{N}\!\left(\mathbf{a}^{(k)} \mid \mathbf{0}, \sigma_a^2 \mathbf{I}_k\right) \mathrm{IG}\!\left(\sigma_a^2 \mid \alpha_a, \beta_a\right) \tag{4.58}$$

$$= \mathrm{IG}\!\left(\sigma_a^2 \mid \alpha_{sa}, \beta_{sa}\right) \tag{4.59}$$

where

$$\alpha_{sa} = \alpha_a + \tfrac{1}{2}k \qquad \text{and} \qquad \beta_{sa} = \beta_a + \tfrac{1}{2}\mathbf{a}^{(k)T}\mathbf{a}^{(k)} \tag{4.60}$$

4.5.1.4   Proposal distribution for changes in model order

All three types of reversible-jump move require a new model order, k', to be proposed. To ensure good mixing, we want most proposed jumps to be small, but occasional large ones to occur too. We choose a discretised Laplacian density, centred on k:

$$J(k' \mid k) \propto \begin{cases} \exp(-\lambda\,|k' - k|) & \text{if } k' \neq k \\ 0 & \text{if } k' = k \end{cases}$$

This distribution has the advantage of being symmetric, so that the $J(k \mid k')/J(k' \mid k)$ part of the acceptance probability is simply unity if k and k' are both within the range $0 \ldots k_{\max}$ and zero otherwise.

Large jumps can be facilitated by thickening the tails of the proposal distribution by using a mixture of a uniform and a Laplacian distribution.

$J(k \mid k)$ can be set to zero, as such a move would have no effect in the schemes of §4.4.1 and §4.4.2 and be equivalent to the Gibbs move of §4.5.1.1 in the scheme of §4.4.3.

4.5.2   Algorithm

Algorithm 4.1 can be used with all three types of reversible-jump move— $\alpha(\cdot)$ is taken from equation (4.28), (4.41) or (4.47) as required. Since the

---

Algorithm 4.1. Reversible-jump sampler for AR model: *for clarity, iteration numbers are omitted.*

Choose initial values
for $i = \{1 \ldots$ number of iterations$\}$
    $k' \sim J(k' \mid k)$
    $\mathbf{a}_u^{(k')} \sim p(\mathbf{a}_u^{(k')} \mid \sigma_a^2)$ *— for moves of §4.4.1 only*
    $z \sim U(0, 1)$
    if $z < \alpha(k \to k' \mid \mathbf{x}, \mathbf{a}^{(k)}, \mathbf{a}^{(k')}, \sigma_a^2, \sigma_e^2)$
        $k = k'$
    end if
    $\mathbf{a}^{(k)} \sim p(\mathbf{a}^{(k)} \mid \mathbf{x}, k, \sigma_a^2, \sigma_e^2)$
    $\sigma_a^2 \sim p(\sigma_a^2 \mid \mathbf{a}^{(k)})$
    $\sigma_e^2 \sim p(\sigma_e^2 \mid \mathbf{x}, k, \mathbf{a}^{(k)})$
end for

---

whole of $\mathbf{a}^{(k)}$ is sampled from its full conditional immediately after each reversible-jump move, there is no need to sample $\mathbf{a}_u^{(k)}$ as part of the move, as the value would be discarded. Sensible initial values for the parameters can be drawn from their prior distributions.

4.6 Results

4.6.1 Synthetic AR data

The rates of convergence of the three methods were compared using signals generated from a synthetic AR(8) process with poles at $0.85 \, e^{\pm j\pi \frac{120}{180}}$, $0.75 \, e^{\pm j\pi \frac{75}{180}}$, $0.75 \, e^{\pm j\pi \frac{45}{180}}$ and $0.8 \, e^{\pm j\pi \frac{15}{180}}$.

For each algorithm, an ensemble of Markov chains was formed by running the sampler 200 times, each time using 4900 samples from a different realisation of the signal and drawing initial values for k, $\mathbf{a}^{(k)}$, σ_a^2 and σ_e^2 from their prior distributions. The hyperparameters were set to $k_{\max} = 20$, $\alpha_a = \beta_a = \alpha_e = \beta_e = 10^{-4}$ and $\lambda = 0.5$. Such parallel runs are not necessary in normal use, but are used here to allow estimation *across* an ensemble of independent realisations of each Markov chain; usually estimation would be performed *along* one chain.

Figure 4.1 shows, for each of the algorithms, at each iteration, a
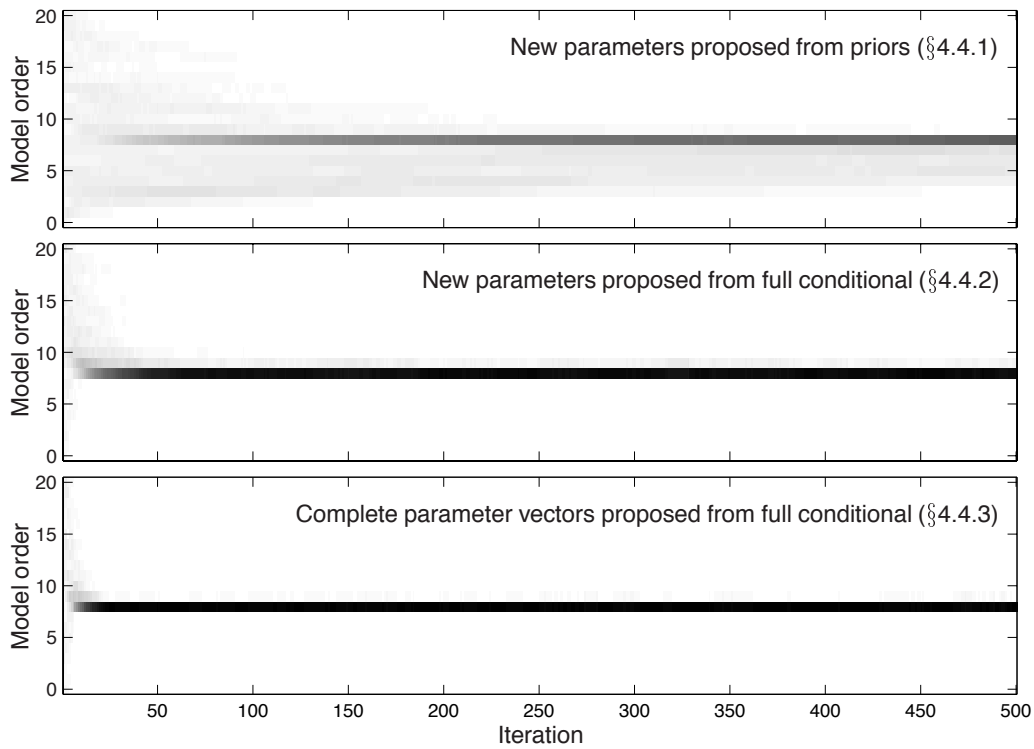
Figure 4.1. Comparison of reversible-jump moves: *evolution of model order histogram—darkness represents frequency of occurrence across the ensemble of 200 runs; the scale is included in Figure 4.2.*
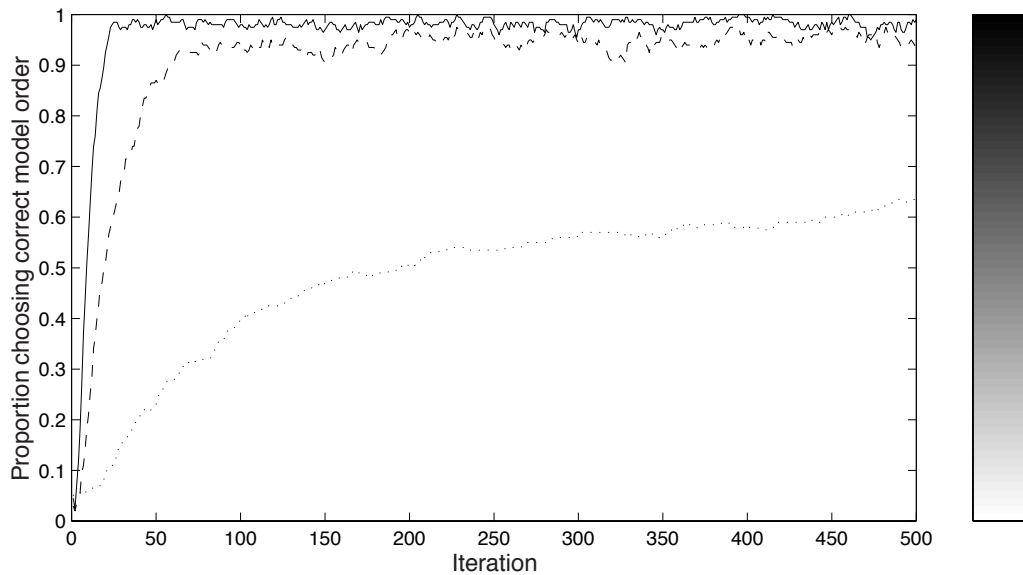
Figure 4.2.   Proportion of ensemble choosing correct order model: *using proposals of §4.4.1 (dotted), §4.4.2 (dashed), §4.4.3 (solid), together with scale for Figure 4.1.*

histogram of the model order across the ensemble. To start with, the histogram is flat, as the initial model order is sampled from the bounded uniform prior. Towards the end of the runs, a significant proportion of the chains are producing models of the correct order. The full parameter vector proposal method (§4.4.3) converges fastest, followed by the method of §4.4.2, followed by that of §4.4.1. Figure 4.2 shows, for each method, the proportion of the runs which choose the correct model order at each iteration. The straightforward method (§4.4.1) performs badly because its acceptance rate is much lower than that of the methods which use the full conditional distributions.

As it converges much faster, we will use the full parameter vector proposal method (§4.4.3) for all further work. For the same data sets, the BIC model selection criterion (§2.1.6.2) consistently suggests an eighth order model and the AIC varies between eighth and ninth order models.

4.6.2 Audio data

To compare this approach to model selection with more conventional techniques, the full parameter vector proposal algorithm (§4.4.3) was used to fit an AR model to a block of 1000 samples from a 44.1 kHz sampled vocal recording, shown in Figure 4.3

The sampler was run for 10 000 iterations, with initial values sampled from the priors. The maximum model order, k_{max}, was set to 120, and the hyperparameters as in the previous experiment. The model order change proposal distribution (§4.5.1.4) was a mixture of a Laplacian with $\lambda = 0.5$ and a uniform distribution over the range $(-k_{max}, k_{max})$, such that each distribution accounted for about half the proposals.

Figure 4.4 shows the sampler output. The initial model order, sampled from the prior, was 112. It can be seen that the sampler very quickly converges to a fairly narrow posterior distribution. Multiple runs, from different starting points, all converged within 100 iterations to the same region.

Monte Carlo estimates of the marginal posterior model order probability distribution, $p(k \mid \mathbf{x})$, are shown in Figure 4.5, along with the AIC and BIC model selection criteria (see §2.1.6.2). The first estimate is based on all iterations except the first 100, which were discarded as burn-in due to the atypical initial values. The second estimate is based on only the first 100 post-burn-in iterations. It can be seen that they agree closely with each other,
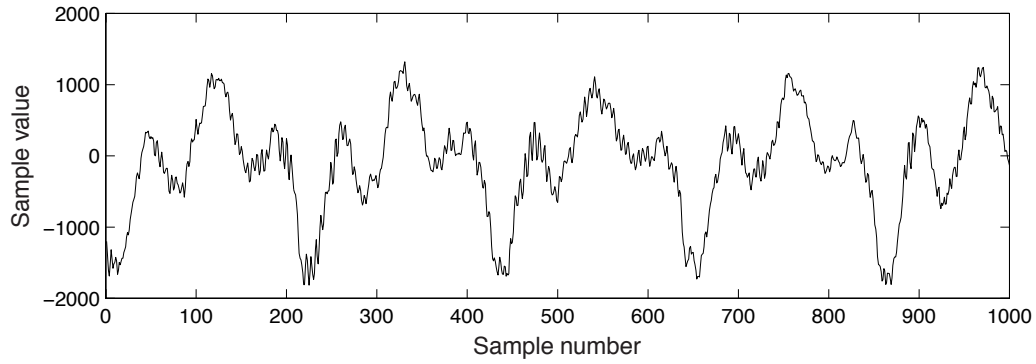
Figure 4.3. Audio signal used for the experiment of §4.6.2.
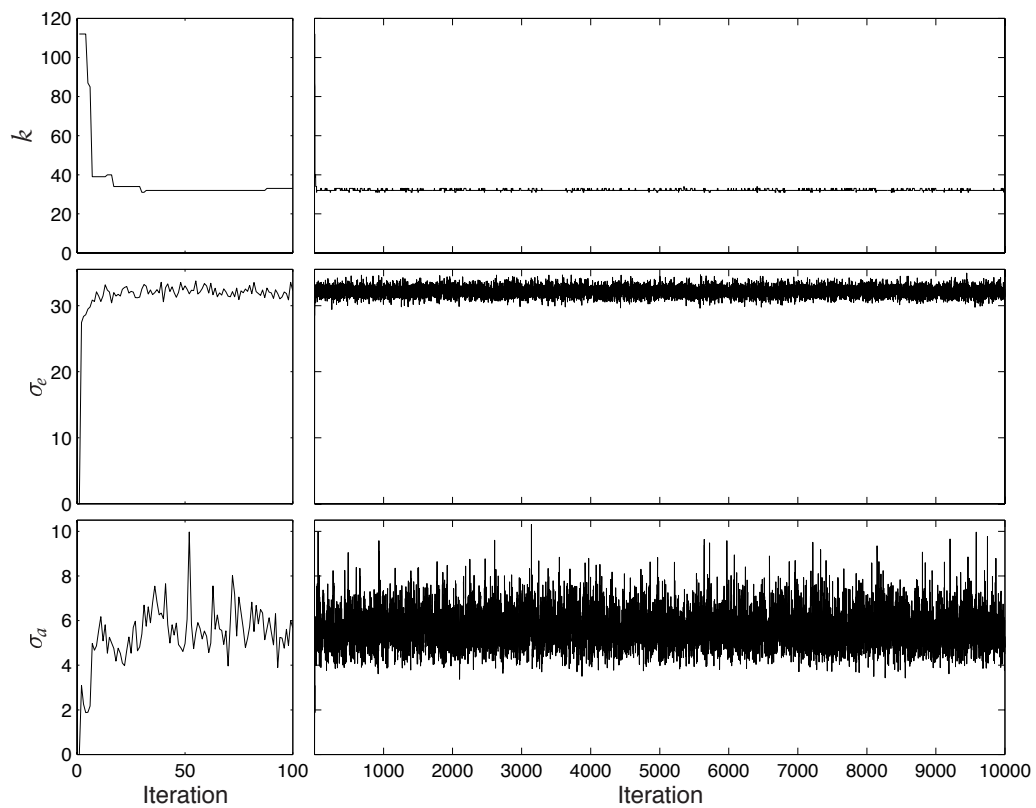
Figure 4.4. Raw reversible-jump sampler output: (from top) *model order, excitation standard deviation and parameter prior standard deviation, with the first 100 iterations shown in more detail.*
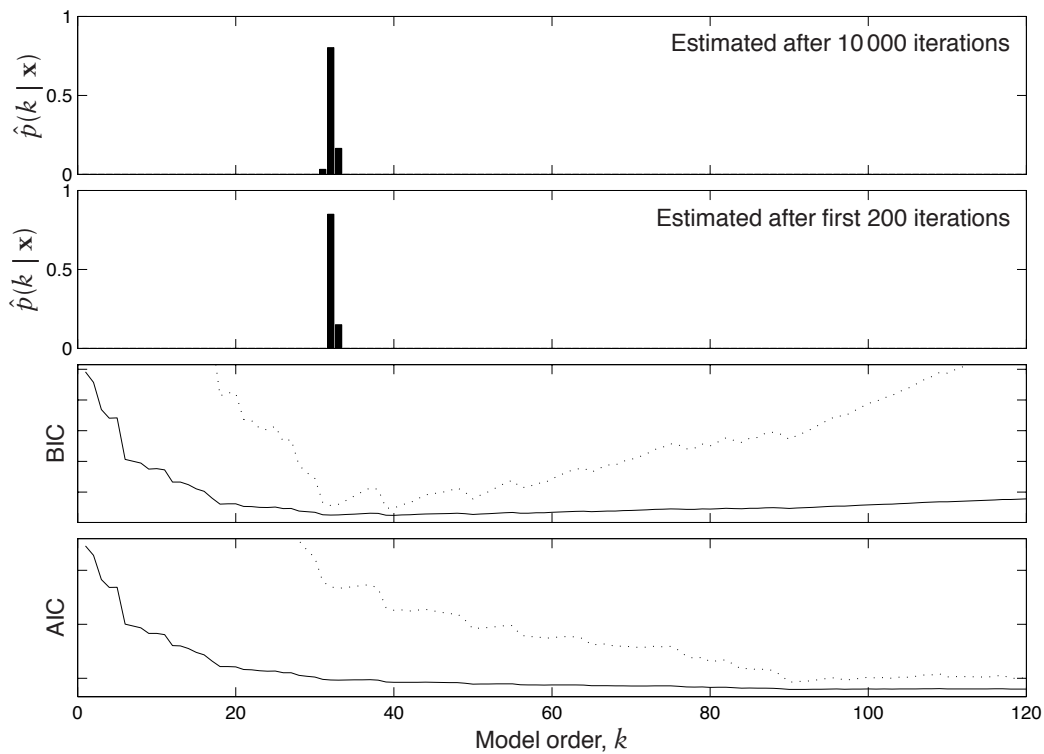
Figure 4.5. AR model order selection: *Monte Carlo estimates of marginal posterior model order probabilities, together with* (solid) *BIC and AIC model order selection criteria and* (dotted) $10\times$ *magnified versions thereof.*

as they did with estimates from separate runs. The *maximum a posteriori* estimate of k is 32, but 33 and 31 also have significant probability. The BIC criterion has a local minimum at 32, but its global minimum is at 40, both of which are reasonable for audio signals. The AIC, which is known to tend to overmodel [112], suggests a model of order 90.

4.7 Application to noise reduction

We now incorporate our new reversible-jump moves into an existing model-based noise reduction algorithm which has previously been used only with fixed model orders.

The removal of white noise, which is perceived as hiss, from audio recordings is a heavily researched area, but model-based methods are not yet widely used.

4.7.1 Frequency domain methods

Most noise reduction algorithms work in the frequency domain (see *e.g.* [5]). The signal is split into blocks within which the frequency components of the signal can be assumed to remain constant; for a sample rate of 44.1 kHz, blocks of length 1024 samples are sufficiently short for this to be reasonable [27]. Each block from the signal is transformed, usually using a windowed *Fast Fourier Transform* (FFT), to the frequency domain. The Fourier coefficients are manipulated in some manner, then the block is inverse transformed back to the time domain. To ensure continuity in the reconstructed signal, overlapping blocks are used, windowed such that they have zero amplitude at the beginning and end. The variations in gain due to the windowing in the FFT and the blocking can be removed by multiplying by a gain compensation window [87].

In the frequency domain, the signal's energy will tend to be concentrated in just a few of the FFT's bins, whereas white noise will be spread evenly across all the bins. Hence applying an *attenuation rule*, which scales up the high amplitude values and scales down the low ones, should increase the signal to noise ratio. A wide range of attenuation rules are used in practice (see [87, §6.1] for a brief overview), most of which are nonlinear, with a threshold at the level of the estimated noise floor.

4.7.2 Musical noise

Musical noise is a common artefact of noise reduction processes. It occurs due to the difficulty in distinguishing between features of a signal which are important and those which are merely components of the random noise. In FFT-based noise reduction, when a frequency bin which does not contain a significant proportion of signal randomly (due to noise) attains a amplitude above the threshold, and hence is not attenuated, it sounds like a short burst of a musical tone in the restored signal. Increasing the threshold will prevent this, but at the cost of further distorting the signal by attenuating perceptually significant components.

Musical noise components differ from block to block, whereas signal components tend to persist for several blocks. Hence a simple measure to reduce musical noise is to median filter the proposed levels of attenuation
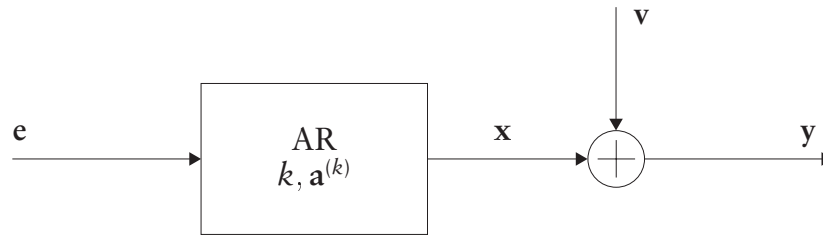
Figure 4.6. Modelling of noisy audio: y *is the observed noisy signal,* **x** *is the noise-free audio signal we wish to estimate,* **v** *is the additive white Gaussian noise component, and* **e** *is the excitation process.*

for adjacent blocks—a component which appears in only a single block will then have no effect on the applied attenuation.

4.7.3   Model formulation

As shown in Figure 4.6, we model the noisy signal as the sum of an AR process, representing the noise-free audio, and a white Gaussian noise process (see *e.g.* [88, 124, 165]). This can be represented in state-space form (see §2.3.3.10) as

$$x_t = \mathbf{a}^{(k)T}\mathbf{x}_{\langle t-k\ldots t-1\rangle} + e_t \qquad\qquad e_t \sim \mathrm{N}\big(e_t \mid 0, \sigma_e^2\big) \qquad\qquad (4.61)$$

$$y_t = x_t + v_t \qquad\qquad v_t \sim \mathrm{N}\big(v_t \mid 0, \sigma_v^2\big) \qquad\qquad (4.62)$$

where equation (4.61) is the *state equation* and equation (4.62) is the *observation equation* and $\mathbf{x}_{\langle i\ldots j\rangle}$ means the ith to jth elements from **x**.

   We assume that the noise variance, σ_v^2, is known. In practical FFT-based noise reduction, it is usually either adjusted by the user, to optimise the trade-off between perceived signal distortion and musical noise, or estimated from a part of the recording which contains only noise.

4.7.4   Simulation smoother

The *simulation smoother* (see *e.g.* [30, 46, 57]) is a method which allows samples to be drawn efficiently from the posterior distribution of the states in a state-space model. In this model, the sequence of states, **x**, defined in equation (4.61) form the desired audio signal.

The task of jointly sampling all of $\mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y}, k, \sigma_a^2, \sigma_v^2)$ is broken down through recursion into two passes through the signal. The first, the forward pass, uses a Kalman filter to estimate the prediction and update distributions, $p(x_{t+1} \mid \mathbf{y}_{\langle 1...t \rangle})$ and $p(x_{t+1} \mid \mathbf{y}_{\langle 1...t+1 \rangle})$. The reverse pass then draws samples $x_t \sim p(x_t \mid \mathbf{x}_{\langle t+1...n_x \rangle}, \mathbf{y})$, starting at the end of the signal. Since, by the probability chain rule,

$$p(\mathbf{x} \mid \mathbf{y}) = \prod_{t=1}^{n_x} p(x_t \mid \mathbf{x}_{\langle t+1...n_x \rangle}, \mathbf{y}) \tag{4.63}$$

this is equivalent to jointly sampling all of $\mathbf{x}$ as required. The simulation smoothing procedure used here is a direct implementation of that described by de Jong & Shephard [46].

4.7.5 Blocking

As with frequency-domain methods (§4.7.1), the AR model can only be assumed to remain stationary over relatively short blocks. There is no need, however, to use overlapping blocks to ensure continuity across block boundaries: this can be done by using the final k values from the previous block as the initial values, $\mathbf{x}_0$, on which probability distributions for the current block are conditioned. Thus we perform *joint* processing of the whole signal. This contrasts with the approach of Lim & Oppenheim [125], in which each block is processed independently.

4.7.6 Algorithm

Algorithm 4.2 shows the algorithm used for the noise reduction experiments. Since the simulation smoother step requires much more computation than all the other sampling steps, several reversible-jump moves are proposed each iteration. To avoid clumsy notation, the algorithm refers only to one block of data; in practice, each block is sampled in turn in each iteration.

4.7.7 Experiments & discussion

The signal **winner** (Track 1 on the accompanying CD—see Appendix C) is a five second extract from a commercial vocal music recording. White Gaussian noise was added at a level 28 dB below the r.m.s. level of the whole signal. Track 2 is the resulting noisy signal.

Algorithm 4.2. Noise reduction for an AR signal of unknown model order:
using the simulation smoother and reversible-jump model selection

---

Choose initial values
for $i = \{1 \ldots$ number of iterations$\}$
$\quad \mathbf{x} \sim p(\mathbf{x} \mid \mathbf{y}, k, \mathbf{a}^{(k)}, \sigma_e^2, \sigma_v^2)$
$\quad$ **for** $r = \{1 \ldots$ number of reversible-jump moves per iteration$\}$
$\quad\quad k' \sim J(k' \mid k)$
$\quad\quad z \sim U(0, 1)$
$\quad\quad$ **if** $z < \alpha(k \rightarrow k' \mid \mathbf{x}, \sigma_a^2, \sigma_e^2)$
$\quad\quad\quad k = k'$
$\quad\quad$ **end if**
$\quad\quad \mathbf{a}^{(k)} \sim p(\mathbf{a}^{(k)} \mid \mathbf{x}, k, \sigma_a^2, \sigma_e^2)$
$\quad\quad \sigma_a^2 \sim p(\sigma_a^2 \mid \mathbf{a}^{(k)})$
$\quad\quad \sigma_e^2 \sim p(\sigma_e^2 \mid \mathbf{x}, k, \mathbf{a}^{(k)})$
$\quad$ **end for**
end for

As a benchmark, the signal was processed by using the same simulation smoother, but with a fixed AR model order of 30. The algorithm was run for 100 iterations, and a Monte Carlo estimate of the signal (Track 3) produced from the final fifty. The noise level was reduced by an average of 3.4 dB (r.m.s.). The reduction in the noise level is audible, but there are disturbing short-duration tones in the quiet parts of the signal, similar to musical noise. For comparison, Track 5 is the same noisy signal processed by a simple FFT-based spectral subtraction algorithm. Noise is reduced by 2.4 dB (r.m.s.), and the musical noise artefacts are of similar intrusiveness.

The same signal was then processed using Algorithm 4.2 for 100 iterations, and the signal again estimated from the final 50 iterations (Track 4). Figure 4.7 shows the signal along with estimates of the posterior distributions of the model order and excitation variance in each block and the noise levels before and after restoration. Although the average level of noise reduction is again 3.4 dB (r.m.s.), similar to that obtained in the fixed model order experiment, no musical noise artefacts can be heard in the restored signal.

It seems likely that the musical noise artefacts in Track 3 occur when the correct model order is smaller than the fixed model order, so some aspects of the noise are incorporated into the model. Incorporating reversible-jump model order selection prevents this.
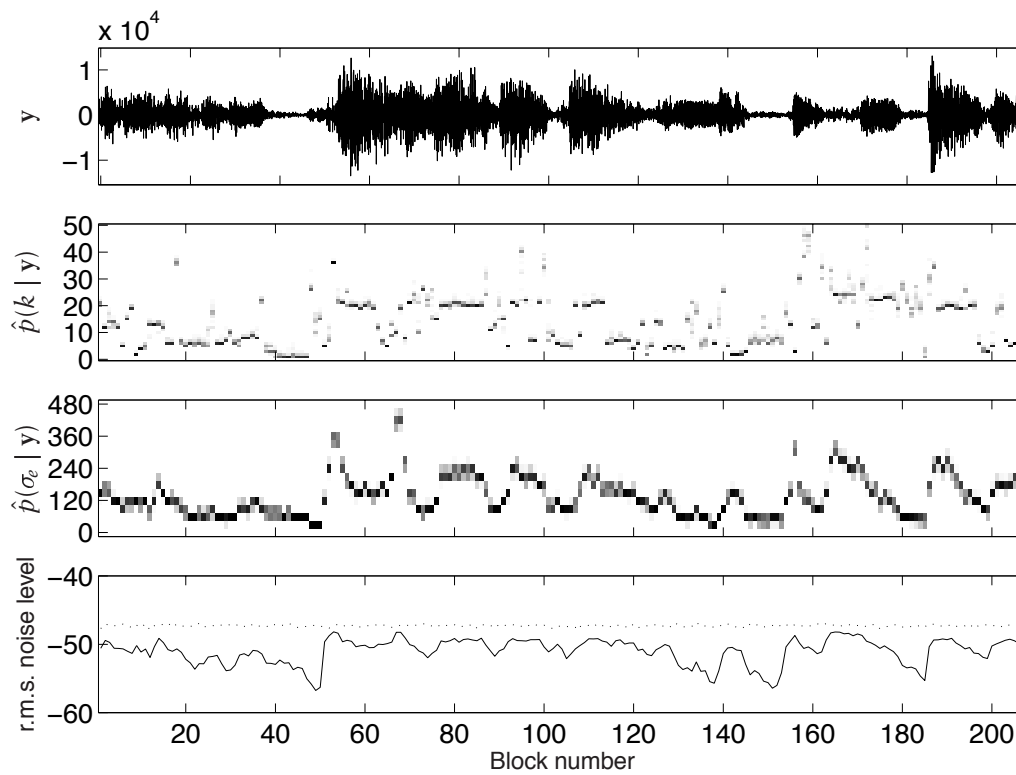
Figure 4.7. Noise reduction in winner: (from top) *signal; estimated posterior model order distribution; estimated posterior excitation standard deviation distribution; noise level before* (dotted) *and after* (solid) *restoration.*
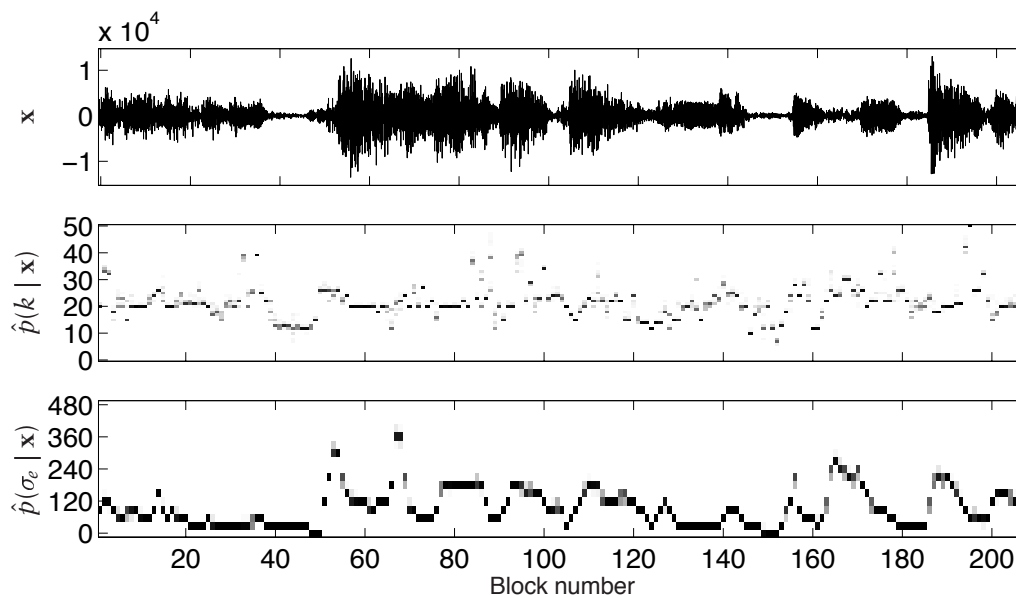
Figure 4.8. Analysis of clean winner signal: (from top) *signal; estimated posterior model order distribution; estimated posterior excitation standard deviation distribution.*

It is not perfect, however: Figure 4.8 shows estimates of the model order and excitation variance produced by using the same reversible-jump algorithm (without the simulation smoothing step) on the clean signal. For example, in the reverberation tail around block 40, estimates from the clean signal suggest a model order around 13, whereas Figure 4.7 suggests orders around 2. This is likely to be because the structure of the quiet signal has been completely swamped by noise. Later in the signal (blocks 100–140), the estimates from the noisy signal repeatedly changes between high values, agreeing with Figure 4.8, and underestimates. This switching behaviour may be the cause of the fluttering noises which are occasionally audible in the restored signal. This could possibly be overcome by introducing, through the prior on k, Markovian dependence (§3.2.2) between the model orders in adjacent blocks, with significant probability assigned to smaller changes in model order. This approach can be extended to the priors for the parameter values and excitation variance (see *e.g.* [54, 165]).

4.8   *Discussion*

In this chapter, we have discussed approaches to AR model order uncertainty using reversible-jump MCMC methods. We then developed new reversible-jump techniques using the natural parameterisation of the AR model, and saw how this approach allows convergence to be greatly speeded by exploiting the analytic properties of the AR model. Proposing full parameter vectors might be expected to be slower, as it does not take advantage of the nested structure of the model, but was instead found to give much better mixing. The MCMC approach readily allows model mixing in applications where the model order is uncertain.

Using the example of model-based audio noise reduction, we have seen how incorporating model order selection into an established MCMC algorithm can improve its performance by avoiding the artefacts which result from overmodelling.

Subset selection in
nonlinear time series models

<div style="text-align: right">5</div>

5.1 Motivation

In Chapter 4, we addressed the situation in which a particular form is assumed for a model, but the number of terms to include is an unknown. The approach developed in Chapter 4 is highly suitable for linear models, in which model terms have a natural ordering, with higher order terms tending to be less significant than those of lower orders.

This is not always the case: particularly in nonlinear models, even if the kth term is highly significant, the preceding terms may not be. Including all these preceding terms could lead to overmodelling.

This problem can become extreme in models whose terms derive from Volterra expansions. If a fifth degree term involving a lag of four samples is required to model a system, then a complete expansion up to this term would contain some 3000 terms.[1] It has been found in various applications [21, 39] that of the order of 10 terms are sufficient to model even highly nonlinear systems. As discussed in Chapter 2, including a large number of redundant terms will lead to overmodelling, as a model with so many parameters is capable of reproducing features of a limited dataset which should be attributed to noise.

Hence there is a need to perform model selection in which each possible *subset* of the available model terms is a candidate model. A subset can be represented by a vector of binary indicators, β, where each element corresponds to one model term, and only those terms for which $\beta_i = 1$ are

---

[1]The number of terms in a triangularised Volterra expansion of degree p and maximum lag k is [143]

$$N_{\text{terms}} = \frac{(p + k)!}{p!\, k!} \tag{5.1}$$

included in the model. The *maximum a posteriori* subset model is then[2]

$$\hat{\beta}_{\mathrm{MAP}} = \arg\max_{\beta}\big(p(\beta \mid \mathbf{y})\big) \qquad (5.2)$$

The task of subset selection in this nonlinear model is similar to that of variable selection in standard regression modelling.

5.2   *Deterministic search methods*

Maximising the posterior probability in equation (5.2) is essentially a multivariate optimisation problem, which is a widely studied field (see *e.g.* [56]). What is different here is that all the variables are binary. This renders concepts such as *steepest descent* of little use.

There are a number of different approaches. Ideally, we would like to find the one subset which forms the optimal model, according to some model selection criterion. This proves to require far too much computation for large models, so we go on to examine a range of suboptimal search techniques.

5.2.1   Exhaustive search

The most straightforward optimal subset selection method is to evaluate the posterior model probability, $p(\beta \mid \mathbf{y})$, or some other model selection criterion (§2.1.6.2) for each of the possible subsets, then pick the one which scores best.

For a model with P candidate terms, there are 2^P possible combinations. With increasing P, this rapidly becomes impractical: for the 3000 term example from the previous example, there are some 10^{900} combinations. This is a very large number.

Fast matrix updating schemes (see *e.g.* [68]) and careful ordering of the models in the search (*e.g.* Gray code ordering, such that only one term changes at each step [155]) can greatly reduce the computation required for an exhaustive search. This allows searches of spaces a few orders of magnitude larger than practicable when using a naïve approach.

---

[2]In this chapter, we denote the data as $\mathbf{y}$, for reasons which will become apparent in Chapter 6.

A possible simplification is to restrict the search to subsets containing, say, 10 or fewer terms. This greatly reduces the search space, to 10^{28} combinations. To put this in context, this space is similar to that which would have to be searched to crack a 94 bit cryptographic key by brute force, which is well beyond the capabilities of current and foreseeable conventional computers.

If we accept that it is too difficult to find the optimal subset, and that we are just looking for an acceptably good model, then the search space can be further reduced by removing models which are *similar* to those already included in the search space. For example, only even numbered lags could be considered, or only subsets which differ in more than one term. This *coarse-grid* search could be followed by a *fine-grid* search to find whether any of the neighbours of the chosen model outperform it [141, §5.5.4].

Even with these rather arbitrary restrictions, the search space is often too large for an exhaustive search to be feasible. The coarseness of the grid is limited by the need for the posterior model probability surface to be smooth on the scale of the coarse grid.

5.2.2   Tree-searching algorithms

Tree searches or *branch and bound* methods (see *e.g.* [101]), can be used to find the optimum subset of each size $(1 \ldots P)$, in terms of *residual sum of squares* (RSS), without testing all possible subsets.

As shown in Figure 5.1, a tree is built, rooted on the full model, with recursive branches, each of which represents exclusion of one term. The branching rules are such that each subset can appear only once. Since the RSS can only increase as a branch is followed to smaller models, branches can be skipped if their RSS has already been beaten by subsets of all the sizes they would contain.

Such a tree search can be used to draw up a shortlist of subsets, one of each size. Since model selection criteria such as the AIC and BIC (§2.1.6.2) are, for models with a given number of parameters, monotonic functions of the RSS, the chosen criterion need only be applied to the shortlist to find the optimum model.

Furnival & Wilson [58] describe a highly optimised algorithm which reuses expensive calculations, such as matrix inversions, wherever possible.

The number of subsets for which we need to estimate the parameters
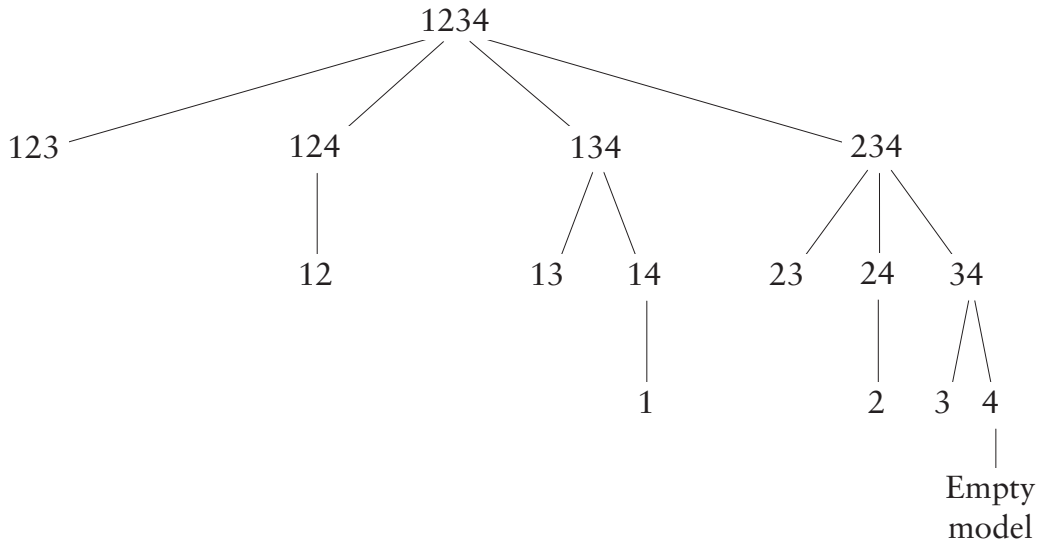
Figure 5.1. Inverse tree for subset selection from four candidate terms: *digits represent model terms included at that node. (Adapted from [58].)*

and evaluate the RSS depends on the search algorithm and where the optimum subsets of each size happen to fall within the tree. In the worst case, the full tree, with 2^P nodes, would need to be evaluated. In the best case, as few as $\frac{1}{2}P^2 + \frac{1}{2}P + 1$ evaluations might be needed. The number of calculations required in a typical search is approximately proportional to $\sqrt{2}^P$ [58]. Clearly, for the relatively large values of P we are considering, this is impractical.

5.2.3   Stepwise algorithms

Forward selection and *backward elimination* are much faster ($O(P^2)$) suboptimal search methods. Rather than searching all possible models, the former starts with a very small model and repeatedly finds the best single term to incorporate into it to improve its performance (as measured by any of the techniques in §2.1.6), until none of the remaining parameters will give a significant improvement. The latter method starts with a full model, and repeatedly selects terms to remove.

    These methods cannot recover once they have chosen an inappropriate term to include or exclude. Pope & Rayner [152] combine these approaches to give the *greatest stepwise improvement* (GSI) algorithm, which considers the effect on the Bayesian evidence of including or excluding each candidate

term, and chooses the inclusion or exclusion which gives the greatest improvement. When none of these choices leads to an improvement, it stops.

The problem with stepwise methods is that it is easy for them to become stuck in local minima. For example, where several candidate terms fulfil a similar role, once one which works adequately has been included, it will not be dropped. However, its presence will make the other similar terms, which may be better, seem redundant, so they will not be tried. Similarly, if a group of terms only functions well when considered together (common in the case of Taylor expansions), then they will not be chosen when the algorithm considers them one by one. Stark [176, §5.2] calls these problems *inclusivism* and *exclusivism*. The former could perhaps be overcome, at the expense of speed, by incorporating $O(P^2)$ "add one and remove one" steps into the GSI algorithm. The latter could be addressed by allowing "add the best q new terms" steps, but these would be expensive: $O(P^q)$.

5.3   Stochastic search methods

We have seen that optimal methods become impractical when the search space is large. We then considered suboptimal search methods, and observed that they tend to get stuck at local minima, as they only allow moves which immediately improve the model.

This can be avoided by introducing an element of randomness, so that it is possible for the algorithm to move past the *energy barrier*[3] separating a local minimum from the global one.

5.3.1   Genetic algorithms

Genetic algorithms, also known as *evolutionary computation*, are a further, non-MCMC, class of algorithms for searching the model space. Introduced in the 1970s [102], and recently applied in fields as diverse as timetabling and DNA sequence alignment [100], they are based on an analogy with biological inheritance and Darwinian evolution.

From a population of randomly-generated candidate models, a new

---

[3]This term is used as an analogy to activation energy barriers in exothermic chemical reactions: in order to get to a lower energy (higher probability) state, it may be necessary to pass through a higher energy (lower probability) one.

generation is bred. The more *fit* a candidate model, the more likely it is to contribute to the next generation. These contributions can be through simple duplication, *mutated* copies of the model, in which some random changes are made, or through *crossover*, in which a new model is generated by randomly intermixing terms from a pair of models. As the process is repeated, the average fitness of each successive generation will tend to increase.

There is great scope for flexibility in choosing the fitness measure, the propagation rules, the population size, and the mutation and crossover operators. If the posterior model probability is used to measure fitness, genetic algorithms can be used for Bayesian model selection.

As the evolution of the population in a genetic algorithm is a Markov process, *i.e.* it depends only on the last state, genetic algorithms can be modelled as Markov chains (see *e.g.* [167]).

5.3.2   Previous MCMC approaches

Taking an MCMC approach to the subset selection process not only introduces randomness, allowing local minima to be escaped, but also has the advantage of allowing model mixing (§2.1.7) and the incorporation of model selection into a larger MCMC framework for solving complex problems.

One way to do this is to construct a Markov chain which moves around the model space by sampling the indicator variables, $\{\beta_i\}$, as well as the other parameters, to produce a sequence of states $\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \ldots$. Once the sequence has converged, it produces dependent samples from the posterior $p(\boldsymbol{\beta} \mid \mathbf{y})$. The values of $\boldsymbol{\beta}$ which occur with highest frequency correspond to the most promising sets of terms to include.

George & McCulloch [66, §3] argue that if we are only interested in the highest probability subsets, rather than the evaluation of the full posterior, a run of much shorter than 2^P iterations should suffice—those areas which have not been visited are of low probability and hence not of interest. This is not an entirely convincing argument, as if we have not visited parts of the posterior, we know nothing about them. There may, for example, be a sharp peak in probability entirely contained within the unvisited area. Nevertheless, for at least mildly well behaved posteriors, the argument seems reasonable.

There are two basic ways in which the binary indicators, $\{\beta_i\}$, can be

incorporated into the model:

- A fixed model can be used, incorporating all candidate terms, with hierarchical mixture priors on the parameter priors, which disable terms by forcing their parameters to small values. The indicators can determine which element of the mixture applies to a term.

- A nested model can be used, with conventional parameter priors. The indicators can then act directly, removing unwanted terms from the model completely.

We now consider previous applications of these two approaches. Godsill [81] presents a generalised model selection framework, of which all of these approaches can be considered to be special cases.

5.3.2.1 Indicators acting on prior

George & McCulloch [66, 67, 68] develop a technique they call *stochastic search variable selection* (SSVS), first for normal linear regression and then extended to generalised linear regression problems. The prior on each parameter b_i is conditional on the associated indicator β_i:

$$p(b_i \mid \beta_i) = (1 - \beta_i)\, N(0, \sigma_c^2) + \beta_i\, N(0, \sigma_b^2) \tag{5.3}$$

where σ_c^2 is very small, and σ_b^2 is the required variance for the prior on b_i. When a term's indicator, β_i, equals zero, the corresponding parameter b_i will be sampled from the posterior corresponding to the use of a very narrow prior (of variance σ_c^2), and hence will tend to be sufficiently close to zero that it does not affect the estimates of the rest of the parameters. But it must also be broad enough for there to be the possibility that the parameter will again be selected, otherwise we have an absorbing state in the Markov chain, which violates convergence conditions [182], a difficulty which is addressed in different ways in the following sections.

Omitting iteration numbers, their sampling scheme is as follows:

$$\mathbf{b} \sim p(\mathbf{b} \mid \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}) \tag{5.4}$$

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta} \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}) \tag{5.5}$$

$$\boldsymbol{\beta} \sim p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{b}, \boldsymbol{\theta}) \tag{5.6}$$

where $\boldsymbol{\theta}$ contains any other parameters common to all subsets; in their example this is just the noise variance. They sample $\mathbf{b}$ as a block and the binary vector $\boldsymbol{\beta}$ component by component in random order:

$$\beta_i \sim p(\beta_i \mid \mathbf{y}, \mathbf{b}, \boldsymbol{\theta}) \tag{5.7}$$

In their example, this probability mass function takes a simple form. Stark [176, §5.4.2] has found sampling the indicator variables in random order to be better than using a fixed sequence.

For problems using conjugate priors, Geweke [71] avoids the compromise in choosing σ_c^2 by sampling each indicator jointly with its associated parameter:

$$(\beta_i, b_i) \sim p(\beta_i, b_i \mid \mathbf{y}, \boldsymbol{\beta}_{[-i]}, \mathbf{b}_{[-i]}, \boldsymbol{\theta}) \tag{5.8}$$

With this sampling scheme, the narrow prior used for disabled terms can be arbitrarily narrow—or even a δ-function at $b_i = 0$—as, once excluded, the term can still be reincluded by this joint move. He also allows for joint priors, $p(\mathbf{b})$.

Godsill & Rayner [86, 88] use a similar approach for outlier detection in noise reduction, in which separate indicators for impulsive noise and excitation outliers are associated with each sample.

5.3.2.2   Indicators acting on model

In the approach of Kuo & Mallick [119, 120], terms are completely removed from the model when their indicators are zero. The values of parameters associated with disabled terms have no effect on the model, so they do not appear in the likelihood. Hence the full conditional posterior distribution from which they are sampled is the same as their prior distribution (§5.5.1.3).

In fact, they can be treated in the same manner as the parameters of unselected models in the composite model space of Carlin & Chib [28] (see §4.2.1), and be drawn from an arbitrary pseudo-prior distribution [81].

Kuo & Mallick [119, 120] sample the parameters as a block and the

indicators one-by-one, conditional on the parameters:

$$\mathbf{b} \sim p(\mathbf{b} \mid \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}) \tag{5.9}$$

$$\beta_i \sim p(\beta_i \mid \mathbf{y}, \boldsymbol{\beta}_{[-i]}, \mathbf{b}, \boldsymbol{\theta}) \tag{5.10}$$

If conjugate priors are used, then joint moves can be made instead [121]:

$$(\beta_i, b_i) \sim p(\beta_i, b_i \mid \mathbf{y}, \boldsymbol{\beta}_{[-i]}, \mathbf{b}_{[-i]}, \boldsymbol{\theta}) \tag{5.11}$$

The parameter values associated with disabled terms then have no effect on the sampling of the indicators, and hence need not be sampled at all— no pseudo-prior distribution need even be chosen. This approach is then exactly equivalent to that of [71].

5.3.3 Subset AR models

The literature we have reviewed above is all concerned with variable selection for linear regression modelling. There has also been some work on subset selection for linear AR models:

- Chen [33] directly applies the SSVS approach (§5.3.2.1) to subset linear AR models.

- Barnett, Kohn & Sheather [12] use a sampling scheme similar to that of Geweke [71], but reparameterise the model in terms of partial autocorrelation coefficients in order to enforce stability. This reparameterisation is not applicable to NAR models.

5.4 Subset selection for Volterra NAR models

We now introduce the Volterra polynomial NAR model for which MCMC sampling schemes for subset selection will be developed in §5.5.

5.4.1 Nonlinear AR model

Using the concepts discussed in §2.3.3, the AR model can be generalised by replacing the weighted sum of past output values with an arbitrary

function [37]:

$$y_t = e_t + f(y_{t-1}, y_{t-2}, \dots, y_{t-\eta_b}) \tag{5.12}$$

where $\{y_t\}$ is the signal, $\{e_t\}$ is i.i.d. Gaussian excitation with variance σ_e^2, and η_b is the maximum lag of the NAR model.

The discrete-time Volterra polynomial expansion [166], a "with memory" extension of the Taylor series, allows the approximation of a broad range of nonlinearities as a simple sum of nonlinear terms (see §2.3.2). The Volterra expansion can be used to approximate the nonlinear function in equation (5.12) by expanding the state space to include a Volterra expansion of past output values:

$$y_t = e_t + \sum_{i=1}^{\eta_b} \sum_{j=1}^{i} b_{(i,j)} \, y_{t-i} \, y_{t-j} + \sum_{i=1}^{\eta_b} \sum_{j=1}^{i} \sum_{k=1}^{j} b_{(i,j,k)} \, y_{t-i} \, y_{t-j} \, y_{t-k}$$

$$+ \text{ higher degree terms} \tag{5.13}$$

where $\{b_{(i,j)}, b_{(i,j,k)}, \dots\}$ are the parameters of the NAR model. To avoid duplication of equivalent polynomial terms, this expansion is in triangular form [143].

5.4.2   Subset & matrix-vector representation

For simplicity, we concatenate the nonlinear parameters of all degrees into a single vector of length n_b,

$$\mathbf{b} = \begin{bmatrix} b_{(1,1)} & b_{(1,2)} & \cdots & b_{(i,j,k)} & \cdots & b_{(\eta_b,\eta_b,\eta_b,\cdots\eta_b)} \end{bmatrix}^T \tag{5.14}$$

As discussed in §5.1, a similar vector of binary indicators, $\boldsymbol{\beta}$, is used to enable and disable terms: if $\beta_i = 1$ then the term with parameter b_i, *i.e.* the ith element in the vector $\mathbf{b}$, is included in the model; otherwise the term is excluded. This use of indicators corresponds to that used by Kuo & Mallick [119] for variable selection (see §5.3.2.2), but was developed independently.

We can extend this to express equation (5.13) in a fully matrix-vector form:

$$\mathbf{e} = \mathbf{y}_1 - \mathbf{Y}(\mathbf{b} \circ \boldsymbol{\beta}) \tag{5.15}$$

where $\circ$ denotes the Hadamard (elementwise) product, $\mathbf{y}_1$ omits the first η_b terms of $\mathbf{y}$, and $\mathbf{Y}$ is a matrix in which row $t - \eta_b$ contains the predictors relating to y_t, *i.e.* the Volterra expansion of $\{y_{t-\eta_b} \ldots y_{t-1}\}$, with the terms in the order that they appear in $\mathbf{b}$. The analogous equation for linear AR models (eq. 2.27) is very similar, but $\mathbf{X}$ has a much simpler structure than $\mathbf{Y}$.

5.4.3   Likelihood

Since the excitation is Gaussian, the approximate likelihood for $\mathbf{y}$ can also be expressed as a multivariate Gaussian:

$$p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_\beta, \sigma_e^2) \approx p(\mathbf{y} \mid \mathbf{y}_0, \boldsymbol{\beta}, \mathbf{b}_\beta, \sigma_e^2) \tag{5.16}$$

$$= \mathbf{N}\big(\mathbf{y}_1 - \mathbf{Y}_\beta \mathbf{b}_\beta \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{5.17}$$

where $\mathbf{y}_0$ is the first η_b elements of $\mathbf{y}$ and the notation $(\cdot)_\beta$ denotes a partition containing only elements corresponding to ones in $\boldsymbol{\beta}$, such that $\mathbf{Y}_\beta \mathbf{b}_\beta = \mathbf{Y}(\mathbf{b} \circ \boldsymbol{\beta})$.

5.4.4   Priors

For the excitation variance, σ_e^2, we use an inverse Gamma prior, as in §4.3.2. For the NAR indicators, we use a simple Bernoulli prior:

$$p(\boldsymbol{\beta}) = \prod_{i=1}^{n_b} \big(\xi \beta_i + (1 - \xi)(1 - \beta_i)\big) \tag{5.18}$$

where the prior probability of inclusion, ξ, is set by the experimenter. Typical values used have been in the range $0.1 \leq \xi \leq 0.5$. Results are not very sensitive to this hyperparameter unless extreme values are used. It is straightforward to use a more informative prior, such as one with different prior probabilities for each term or with dependence between terms, should such knowledge be available.

Independent Gaussian priors are used for the NAR parameters. We expect *a priori* that the values of the NAR parameters of different degrees will be of different magnitudes, so we partition the parameter vector by degree. For the parameters associated with terms of degree m, we use the

prior distribution

$$p(\mathbf{b}_{\{m\}} \mid \sigma_b^2) = \mathbf{N}\left(\mathbf{b}_{\{m\}} \mid \mathbf{0}, \tfrac{\sigma_b^2}{s_{\{m\}}}\mathbf{I}_{n_{b_{\{m\}}}}\right) \tag{5.19}$$

The factor $s_{\{m\}} = E(|y_t^m|)$ scales the prior distribution to have the same effect on parameters from different degrees. This prior can also be treated as a zero-mean multivariate Gaussian with an appropriate diagonal covariance matrix, $\mathbf{C}_{p\mathbf{b}}$. The hyperparameter, σ_b^2, is given an inverse Gamma prior, which is again conjugate. Should prior knowledge justify it, any arbitrary multivariate Gaussian prior could be used instead, without any major changes to the sampling schemes.

5.4.5 Problem formulation

In a stand-alone model selection problem, we want to find the value of β which maximises the marginal posterior distribution,

$$p(\beta \mid \mathbf{y}) = \int \cdots \int_{\mathbf{b}_\beta, \sigma_b^2, \sigma_e^2} p(\beta \mid \mathbf{y}, \mathbf{b}_\beta, \sigma_b^2, \sigma_e^2)\, p(\mathbf{b}_\beta \mid \sigma_b^2)\, p(\sigma_b^2)\, p(\sigma_e^2)\, d\mathbf{b}_\beta\, d\sigma_b^2\, d\sigma_e^2$$

$$\tag{5.20}$$

If our ultimate aim is Bayesian inference about some other quantity—as it would be in interpolation or prediction problems, for example—then we want to perform model mixing (§2.1.7), and thus need to evaluate all parts of the posterior distribution which have significant probability.

5.5   Markov chain Monte Carlo

Since we cannot evaluate equation (5.20) analytically, we take an MCMC approach, as described in §5.3.2.

5.5.1 Sampling strategies

We consider sampling steps which exploit some of the analytic properties of the Volterra NAR model.

5.5.1.1 Joint sampling

If variables are strongly dependent, the Gibbs sampler will tend to converge slowly [182]. Since there is likely to be strong interdependence between the indicator and parameter of each term, we speed convergence by sampling *jointly* from the indicators and their associated parameters, which can be viewed as equivalent to Geweke's [71] approach to variable selection problems:

$$(\beta_u, b_u) \sim p(\beta_u, b_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_f, \sigma_b^2, \sigma_e^2) \tag{5.21}$$

where $(\cdot)_u$ denotes the element which is being *updated* in this move, $(\cdot)_f$ contains the remainder of the elements, which are being treated as *fixed* for this step (§4.4.1.1)[4] Each iteration, this sampling operation is performed once for each term, in a random scan.

   The joint sampling operation of step (5.21) can be performed in two steps using the method of composition [179]:

$$\beta_u \sim p(\beta_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_f, \sigma_b^2, \sigma_e^2) \tag{5.22}$$

$$b_u \sim p(b_u \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\boldsymbol{\beta}_f}, \sigma_b^2, \sigma_e^2) \tag{5.23}$$

where $\mathbf{b}_{\boldsymbol{\beta}_f}$ consists of those elements of $\mathbf{b}_f$ for which the corresponding indicators in $\boldsymbol{\beta}_f$ are set to one. The distributions required for steps (5.22) & (5.23) are derived in §5.5.2.1.

   Note that step (5.22) is *not* conditional on $\mathbf{b}_u$, which is analytically marginalised. As discussed in §5.3.2.2, if, after step (5.22), $\beta_u = 0$ then step (5.23) need not be carried out, as b_u will then have no effect on the model, and no future sampling moves would be conditioned on the sampled value.

5.5.1.2 Blockwise sampling

There will also be interdependence between the parameters and indicators of different terms. We can address this by multivariate sampling of the

---

[4]In this case, $(\cdot)_f$ is equivalent to $(\cdot)_{[-u]}$, but u becomes a set in §5.5.1.2.

indicators, in blocks of size n_u, again jointly with the associated parameters:

$$(\boldsymbol{\beta}_u, \mathbf{b}_u) \sim p(\boldsymbol{\beta}_u, \mathbf{b}_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_f, \sigma_b^2, \sigma_e^2) \qquad (5.24)$$

which again can be performed in two steps:

$$\boldsymbol{\beta}_u \sim p(\boldsymbol{\beta}_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_f, \sigma_b^2, \sigma_e^2) \qquad (5.25)$$

$$\mathbf{b}_{\beta_u} \sim p(\mathbf{b}_{\beta_u} \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \qquad (5.26)$$

These conditional distributions are derived in §5.5.2.1. Step (5.26) only updates the parameters for those terms whose indicators are one, as the values of disabled terms' parameters are never used.

The greatest improvement in mixing will be achieved if the most highly correlated model terms are sampled jointly. In a Volterra expansion, however, it is hard to predict which terms these will be, so different random groups of size n_u are sampled each iteration such that each term is sampled once.

Step (5.25) requires the evaluation of the conditional for 2^{n_u} combinations of terms, so n_u will generally be quite small. Varying n_u allows a trade-off between the number of iterations required for convergence and the computational complexity of each iteration.

5.5.1.3 *Straightforward univariate sampling*

For comparison, a straightforward method is also used, similar to that of Kuo & Mallick [119], discussed in §5.3.2.2. Each indicator is sampled conditional on the value of the corresponding parameter:

$$\beta_u \sim p(\beta_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}, \sigma_e^2) \qquad (5.27)$$

The distribution from which b_u is sampled depends on the state of the indicator. If $\beta_u = 1$ then b_u is drawn from its full conditional distribution, which is the same as step (5.23). If $\beta_u = 0$ then, although b_u has no effect on the model, its value will be used next time the corresponding indicator is updated, so it must be sampled. Any arbitrary pseudo-prior can be used (see §5.3.2.2), but we follow Kuo & Mallick [119, 120] in choosing the

parameter's prior distribution. Hence,

$$
b_u \sim \begin{cases} p(b_u \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) & \text{if } \beta_u = 1 \\ p(b_u \mid \sigma_b^2) & \text{if } \beta_u = 0 \end{cases} \tag{5.28}
$$

The probability mass function for step (5.27) is derived in §5.5.2.2.

5.5.2   Conditional distributions

We now derive the distributions from which we need to sample.

5.5.2.1   Joint, blockwise sampling

We consider here the distributions required for the joint, blockwise sampling scheme of §5.5.1.2. Those required for §5.5.1.1 can be obtained simply by reducing these to one dimension.

We derive the (discrete) distribution for β_u (step 5.25) from the likelihood as follows:

$$
p(\mathbf{y}, \mathbf{b}_{\beta_u} \mid \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) = \overbrace{p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_\beta, \sigma_e^2)}^{\text{Likelihood}} \cdot \overbrace{p(\mathbf{b}_{\beta_u} \mid \sigma_b^2)}^{\text{Prior}} \tag{5.29}
$$

$$
p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) = \int p(\mathbf{y}, \mathbf{b}_{\beta_u} \mid \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \, d\mathbf{b}_{\beta_u} \tag{5.30}
$$

$$
p(\beta_u \mid \mathbf{y}, \beta_f, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \propto p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \cdot \underbrace{p(\beta_u)}_{\text{Prior}} \tag{5.31}
$$

which simplifies, as shown in §B.2, to

$$
p(\beta_u \mid \mathbf{y}, \beta_f, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \propto p(\beta_u) \sqrt{\frac{|\mathbf{C}_{cb_{\beta_u}}|}{|\mathbf{C}_{pb_{\beta_u}}|}} \exp\left(\tfrac{1}{2}\boldsymbol{\mu}_{cb_{\beta_u}}^T \mathbf{C}_{cb_{\beta_u}}^{-1} \boldsymbol{\mu}_{cb_{\beta_u}}\right) \tag{5.32}
$$

where

$$
\mathbf{C}_{cb_{\beta_u}}^{-1} = \tfrac{1}{\sigma_e^2}\mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u} + \mathbf{C}_{pb_{\beta_u}}^{-1} \tag{5.33}
$$

$$
\boldsymbol{\mu}_{cb_{\beta_u}} = \tfrac{1}{\sigma_e^2}\mathbf{C}_{cb_{\beta_u}} \mathbf{Y}_{\beta_u}^T \mathbf{e}_f \tag{5.34}
$$

where $\mathbf{Y}_{\beta_u}$ contains those columns of $\mathbf{Y}$ which correspond to ones in β_u, and $\mathbf{e}_f = \mathbf{y}_1 - \mathbf{Y}_{\beta_f}\mathbf{b}_{\beta_f}$, the excitation which would need to be applied if β_u

was all-zero. To draw a sample from this discrete, multivariate distribution, equation (5.32) is evaluated for each of the 2^{n_u} possible values of β_u. These values are then used to form a c.d.f. so that the method of §3.4.1 can be used.

Step (5.26) is a straightforward draw from a multivariate Gaussian:

$$p(\mathbf{b}_{\beta_u} \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \propto \mathbf{N}\big(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{cb_{\beta_u}}, \mathbf{C}_{cb_{\beta_u}}\big) \tag{5.35}$$

5.5.2.2 Univariate sampling

In step (5.27), each indicator is sampled conditional on the parameter values. From Bayes' theorem,

$$p(\beta_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}, \sigma_e^2) \propto p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}, \sigma_e^2)\, p(\beta_u) \tag{5.36}$$

which can be written using equation (5.17) as

$$= p(\beta_u)\, \mathbf{N}\big(\mathbf{y}_1 - \mathbf{Y}_\beta \mathbf{b}_\beta \mid 0, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{5.37}$$

which can be rearranged, in a similar manner to §B.2, as

$$= p(\beta_u)\, \mathbf{N}\big(\mathbf{y}_1 - \mathbf{Y}_{\beta_f} \mathbf{b}_{\beta_f} - \mathbf{Y}_u \beta_u \mid 0, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{5.38}$$

$$= p(\beta_u)\, \mathbf{N}\big(\mathbf{Y}_u \beta_u \mid \mathbf{e}_f, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{5.39}$$

where $\mathbf{e}_f$ is defined as in §5.5.2.1. This can then be rearranged using equation (A.5) and greatly simplified by neglecting any terms which are independent of β_u, to give

$$p(\beta_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}, \sigma_e^2) \propto \begin{cases} p(\beta_u)\, \exp\left(-\frac{1}{2\sigma_e^2}\big(b_u^2 \mathbf{Y}_u^T \mathbf{Y}_u - 2b_u \mathbf{Y}_u^T \mathbf{e}_f\big)\right) & \text{if } \beta_u = 1 \\ p(\beta_u) & \text{if } \beta_u = 0 \end{cases} \tag{5.40}$$

The normalising constant can be computed simply by summing over the two possible states.

5.5.2.3 Other sampling steps

It is straightforward to sample all the NAR parameters which are currently included in the model, $\mathbf{b}_\beta$, jointly using equation (5.35) by partitioning such

---

Algorithm 5.1. Subset selection for the NAR model using the Gibbs sampler.

Choose initial values
for $i = \{1 \dots \text{number of iterations}\}$
    repeat
        Choose a group $(\cdot)_u$ of n_u nonlinear model terms to sample
$$\left.\begin{array}{l} \boldsymbol{\beta}_u \sim p(\boldsymbol{\beta}_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_{\boldsymbol{\beta}_f}, \sigma_b^2, \sigma_e^2) \\ \mathbf{b}_{\boldsymbol{\beta}_u} \sim p(\mathbf{b}_{\boldsymbol{\beta}_u} \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\boldsymbol{\beta}_f}, \sigma_b^2, \sigma_e^2) \end{array}\right\} \text{— or the steps of §5.5.1.3}$$
    until all model terms have been sampled
    $\mathbf{b}_{\boldsymbol{\beta}} \sim p(\mathbf{b}_{\boldsymbol{\beta}} \mid \mathbf{y}, \boldsymbol{\beta}, \sigma_b^2, \sigma_e^2)$
    $\sigma_b^2 \sim p(\sigma_b^2 \mid \mathbf{b}_{\boldsymbol{\beta}})$
    $\sigma_e^2 \sim p(\sigma_e^2 \mid \mathbf{y}, \boldsymbol{\beta}, \mathbf{b}_{\boldsymbol{\beta}})$
end for

---

that $(\cdot)_u$ contains all the terms, leaving $(\cdot)_f$ empty. Making this move occasionally can further improve mixing.

The hyperparameter, σ_b^2, and excitation variance, σ_e^2, are sampled from their full conditionals, which are inverse Gamma distributions, in simple Gibbs sampler moves, as described in §§4.5.1.3 & 4.5.1.2.

5.5.3   Algorithm

The sampling steps are carried out as shown in Algorithm 5.1. Initial values could be drawn from the priors, although, for the indicators, this is unlikely to give a better starting point than just starting at any arbitrary subset.

5.6   Results

Two experiments were performed: the first to verify that model selection is performed correctly, and the second to compare the performance of the various sampling schemes.

5.6.1   Verification

The following NAR process was simulated:

$$y_t = e_t - 0.5y_{t-2} + 0.3y_{t-1}^2 - 0.1y_{t-1}\,y_{t-2}^2 \tag{5.41}$$

$$e_t \sim \mathrm{N}\big(e_t \mid 0, 0.25\big) \tag{5.42}$$
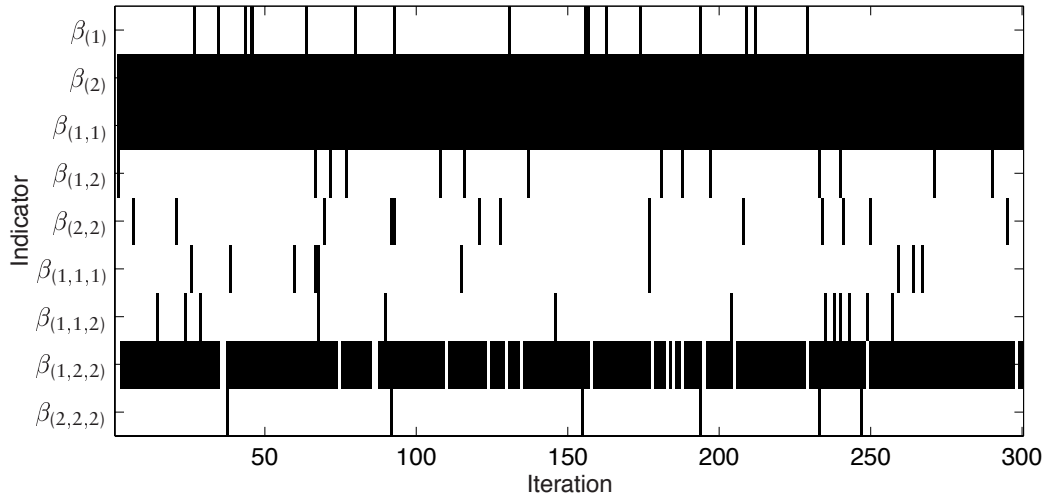
Figure 5.2. Evolution of indicators over first 300 iterations: *black pixels represent $\beta_i = 1$.*
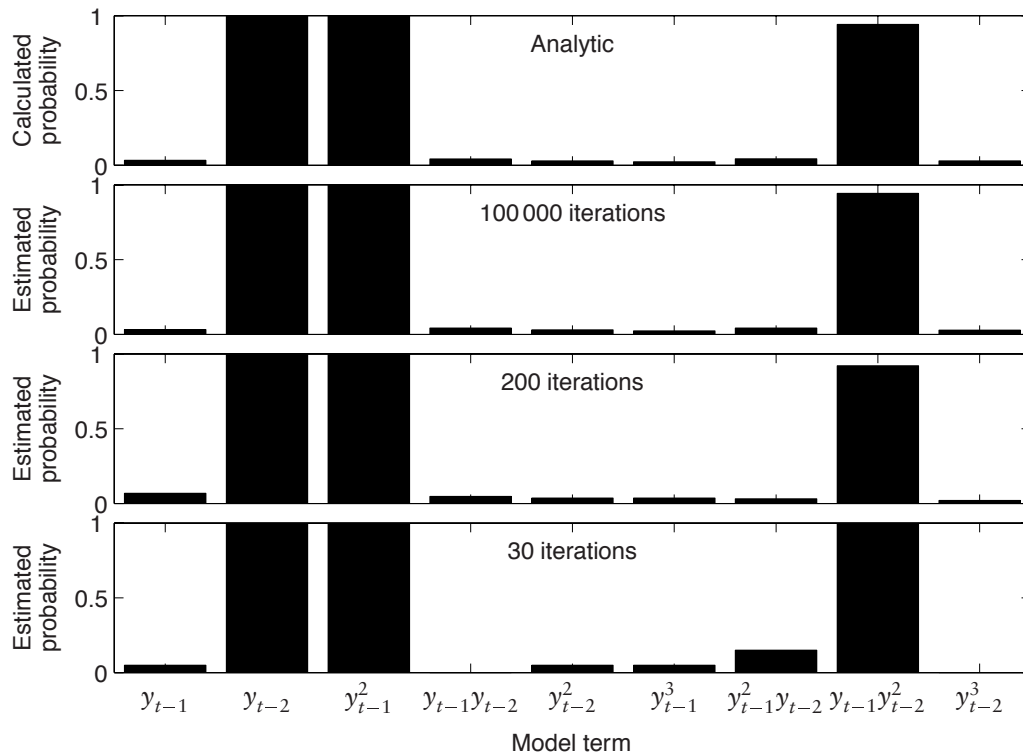
Figure 5.3. Marginal model term posterior probabilities: *analytically calculated probabilities, and estimates based on the first 100 000, 200 or 30 iterations of the Gibbs sampler (after a burn-in of 10 iterations).*

Table 5.1. Composition of subsets in Figure 5.4: *the symbol ● indicates that the term is included in the subset, and ○ that it is not. Subset* a *corresponds to equation* (5.41).

Indicator	Term	Subset											
		a	b	c	d	e	f	g	h	i	j	k	l
$\beta_{(1)}$	y_{t-1}	○	○	○	○	●	○	○	○	●	○	○	○
$\beta_{(2)}$	y_{t-2}	●	●	●	●	●	●	●	●	●	●	●	●
$\beta_{(1,1)}$	y_{t-1}^2	●	●	●	●	●	●	●	●	●	●	●	●
$\beta_{(1,2)}$	$y_{t-1}\,y_{t-2}$	○	○	○	●	○	○	○	○	○	●	○	○
$\beta_{(2,2)}$	y_{t-2}^2	○	○	○	○	○	●	○	○	○	○	○	○
$\beta_{(1,1,1)}$	y_{t-1}^3	○	○	○	○	○	○	○	●	○	○	○	○
$\beta_{(1,1,2)}$	$y_{t-1}^2\,y_{t-2}$	○	○	●	○	○	○	○	○	○	●	●	○
$\beta_{(1,2,2)}$	$y_{t-1}\,y_{t-2}^2$	●	○	●	●	●	●	●	●	○	●	○	○
$\beta_{(2,2,2)}$	y_{t-2}^3	○	○	○	○	○	○	●	○	○	○	○	●
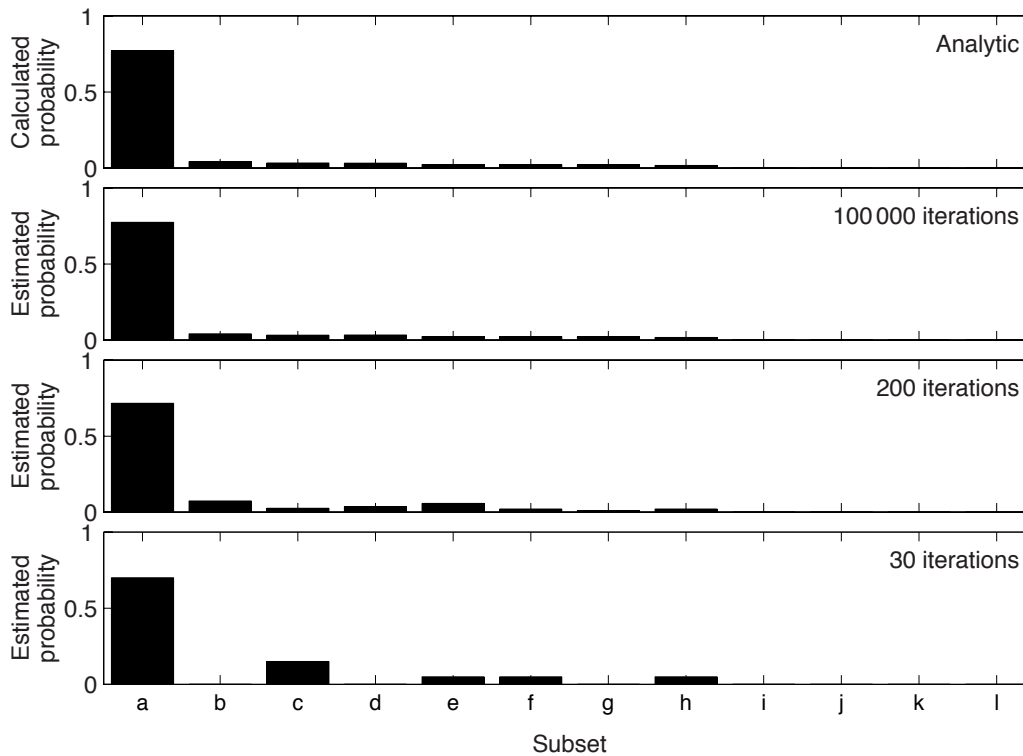
Figure 5.4.   Subset model posterior probabilities: *analytically calculated probabilities, and estimates based on the first 100 000, 200 or 30 iterations of the Gibbs sampler (after a burn-in of 10 iterations). Subsets labelled as in Table 5.1.*

and 1000 samples taken for analysis. All nine first, second and third-degree terms up to lag two were used as candidates.

Starting from an empty model, the sampler was run for 100 000 iterations, sampling indicators in random groups of three elements, β_u, having marginalised $\mathbf{b}_u$. To facilitate comparison between the sampler output and analytically computed posterior model probabilities, σ_e^2 was fixed at the correct value, and σ_b^2 at 0.25.

Figure 5.2 shows the values of the indicators in each of the first 300 iterations. It can be seen that the correct terms are chosen by the third iteration, but that there remains some uncertainty, due to the shortness of the data block.

Figure 5.3 shows the marginal model term posterior probabilities, $p(\beta_i \mid \mathbf{y}, \sigma_b^2, \sigma_e^2)$, which were computed analytically, together with estimates from the sampler output. The first estimate was made using all 100 000 iterations, after discarding the first 10 as burn-in due to the atypical initial state. It agrees extremely closely with the analytic result—the largest error in probability is 0.001. An estimate made from only the first 200 iterations has much greater errors (0.03), but the overall shape is still very close. If only the first 30 iterations are used, the high probability terms are still correctly identified.

A more useful approach is to consider the posterior model probabilities, $p(\beta \mid \mathbf{y}, \sigma_b^2, \sigma_e^2)$. These were computed analytically for all 512 possible subset models. This exhaustive computation is only feasible with such a small number of candidate terms (see §5.2.1). The twelve most probable subsets, which account for 98% of the total probability, are identified in Table 5.1. Figure 5.4 plots these calculated probabilities together with estimates of the same subsets' posterior probabilities made from the sampler output. Again, the estimate from 100 000 iterations is very close, with the largest error being 0.002. After 200 iterations, the basic shape is the same, but there are noticeable errors. After only 30 iterations, the most probable subset is clearly identified, but the tails have not been explored.

5.6.2   Comparison of sampling schemes

In order to compare the sampling schemes listed in Table 5.2, each was run 580 times for 50 iterations, each time on a different realisation of 5000

samples from the NAR process with parameters

$$b_{(2)} = -0.2 \qquad b_{(3)} = 0.1 \qquad b_{(1,2)} = 0.2$$
$$b_{(1,3)} = -0.2 \qquad b_{(2,3)} = 0.2 \qquad b_{(1,2,3)} = -0.3$$

(5.43)

and white Gaussian excitation with variance 0.18. Each run started with an empty model and arbitrary hyperparameter values. All 19 Volterra terms up to third degree, third lag were available as candidates.

Figure 5.5 shows the mean across the ensemble of runs of each indicator at each iteration[5]. When all runs have converged, each should produce samples from the correct posterior distribution, so the ensemble mean of each indicator value should be an estimate of $p(\beta_i \mid \mathbf{y})$, and hence should not change significantly between iterations. It can be seen that all three sampling schemes seem to converge quickly, but scheme N1 takes slightly longer than schemes M1 or M5.

Figure 5.6 shows the same data in a different manner, for easier comparison. The proportion of the runs which choose the correct subset converges, under all sampling schemes, to around 0.26. There is considerable model uncertainty in this problem due to the large number of candidate models and the small amount of data—there are many subset models which differ from the correct model only by the inclusion or exclusion of a small number of terms. If the Hamming distance is used as a measure of model error, it can be seen that the mean distance converges to about 1.5 terms. In both graphs, the lines for schemes M2, M3 and M4 fall between those for M1 and

---

[5]This can be visualised as multiple images similar to Figure 5.2, but more faint, overlaid and held up to a strong light.

Table 5.2. Sampling schemes used in comparison experiment.

Scheme	Description	
N1	Non-marginalised sampling, in blocks of 1	(§5.5.1.3)
M1	Marginalised sampling, in blocks of 1	(§5.5.1.1)
M2	Marginalised sampling, in blocks of 2	(§5.5.1.2)
M3	Marginalised sampling, in blocks of 3	(§5.5.1.2)
M4	Marginalised sampling, in blocks of 4	(§5.5.1.2)
M5	Marginalised sampling, in blocks of 5	(§5.5.1.2)
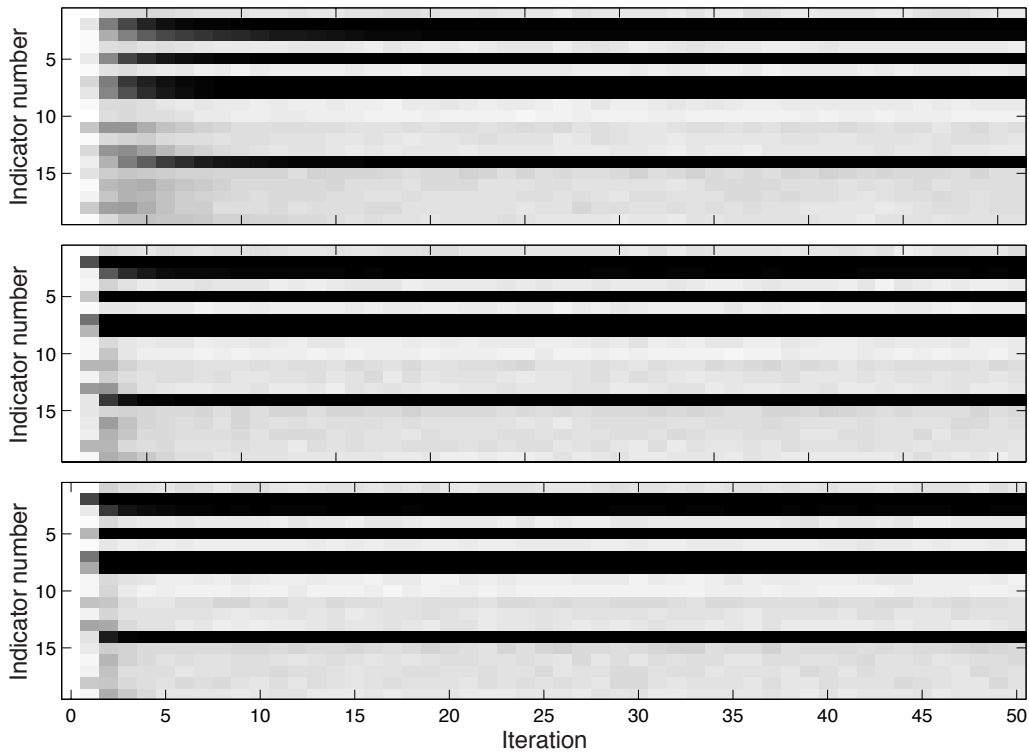
Figure 5.5. Proportion of runs choosing each model term: *under sampling scheme (top) N1, (middle) M1 and (bottom) M5. The correct terms are those numbered 2, 3, 5, 7, 8 and 14. Darker pixels represent higher proportions, following the scale of Figure 4.2.*

M5, but are omitted for clarity.

Clearly, marginalising the parameter values improves the rate of convergence. Blockwise sampling then gives a slight further improvement, although at considerable computational cost.

5.7   *Discussion*

In this chapter, we have considered the need for subset selection in nonlinear models and found that exhaustive searches are infeasible, other than for small models, and that stepwise searches can have problems with local minima.

We have discussed genetic algorithms and existing MCMC approaches to subset selection, then introduced a related MCMC method applied to our model and demonstrated how its performance can be improved by exploit-
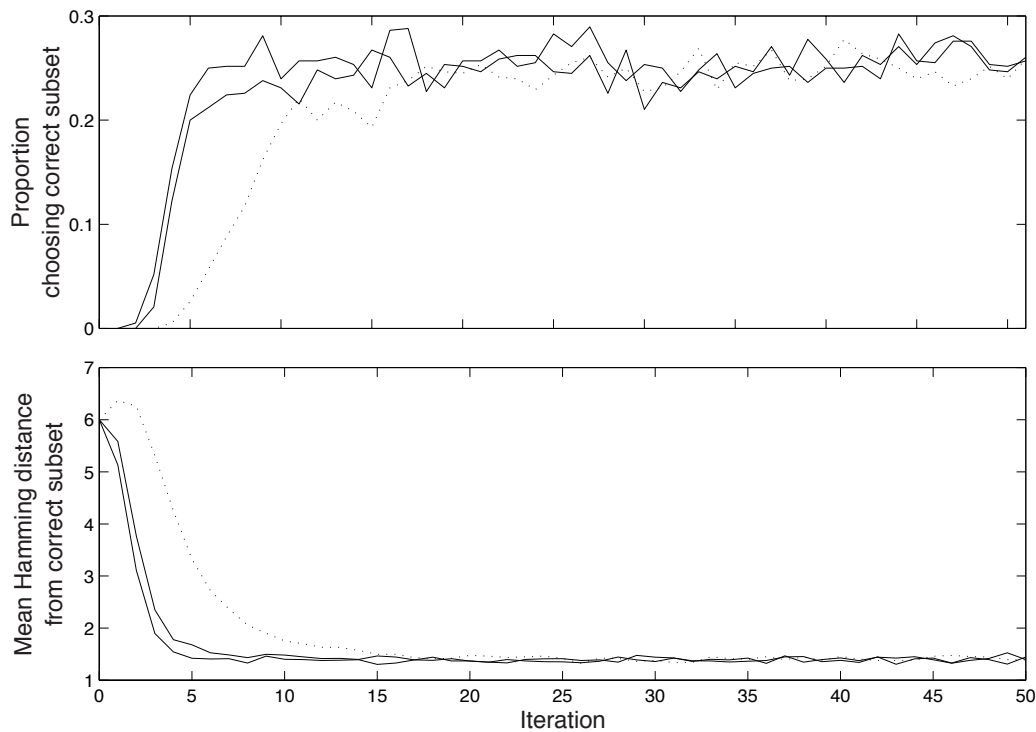
Figure 5.6. Measures of convergence to correct subset: *for sampling scheme* (dotted) *N1,* (light) *M1 and* (heavy) *M5.*

ing the analytic properties of the polynomial NAR model.

By applying the MCMC method to a problem small enough for all posterior probabilities to be computed analytically, we have shown that it produces very good estimates of model probabilities from a long run, and that the most probable models are correctly identified even in a very short run.

Where this sort of problem occurs in models with a natural ordering, for example an MA model of an echo path, it would be best addressed with an approach which favours choosing contiguous blocks of terms. This could be done through the use of a more elaborate prior on the subset indicators, or alternatively through the use of a reversible-jump sampler with specialised moves for splitting and merging blocks of relevant terms.

Restoration of nonlinearly distorted audio

<div style="text-align: right; font-size: 2em;">6</div>

6.1   Introduction

As discussed in §2.2.3, autoregressive processes are used to model a wide range of signals. This chapter addresses the problem of reconstructing such a signal from a nonlinearly distorted version of it, when neither the precise form of the nonlinearity nor the order of the AR process are known.

6.1.1   Nonlinear distortion

Most research into audio restoration techniques has concentrated on the removal of clicks, crackle and surface noise (see *e.g.* [45, 87, 111, 198]), all of which are effectively separate signals which are independent of, and additively superimposed on, the desired clean signal. The restoration of signals which have been degraded by passing through a nonlinear channel has received relatively little attention.

Under ideal conditions, passing an audio signal through an analogue recording stage introduces little nonlinear distortion. It is, however, a considerable problem with many archived recordings. Some of the main causes are:

- Saturation in magnetic recording (see *e.g.* [53, 103])

- Tracing distortion [43, 123] (before precompensation was introduced [159, 203]) and groove deformation [11, 174] in records

- The inherent nonlinearity of variable density optical soundtracks [4]

6.1.2 Memoryless nonlinearities

The intended application is the restoration of nonlinearly distorted audio signals. Most previous work on this problem has considered only memoryless nonlinearities.

6.1.2.1 *Histogram equalisation*

Histogram equalisation [207] is a simple technique to estimate a memoryless nonlinear transfer function through which a speech signal has been passed. A smooth function is fitted through a histogram of sample values from an extract of the signal. This is compared with a reference histogram shape, based on analysis of a range of speakers, and a 1:1 mapping is derived which will make the smoothed histogram conform with the reference one. This mapping is then applied to the distorted signal.

Because it assumes that the original signal closely conforms to a standard reference histogram, this method cannot readily be applied to complex music signals, where histograms differ greatly between recordings and vary significantly over the duration of a recording. The algorithm was originally proposed for use in speech communication channels, and has led to a patented device [206]. A related method has been used to restore recordings made using early analogue-to-digital convertors with non-uniform quantisation step heights and some missed codes [196]. Since these are all small-scale, local defects, they can be reduced by smoothing the histogram, without the need for a reference.

6.1.2.2 *Signal reconstruction with known nonlinearity*

For situations in which distortion is caused by a known memoryless nonlinearity, an iterative algorithm [154] has been proposed to reconstruct the original signal where only a bandlimited version of the distorted signal is available. The algorithm does not appear to have been developed further, possibly because it is unusual for the exact form of the nonlinearity to be known.

6.1.2.3 *Model-based estimation of nonlinear function*

Mercer [141, Ch. 3,4] uses an AR model for the signal, and a Taylor expansion to model the memoryless nonlinearity. Although the parameter

estimation procedure works correctly for synthetically distorted audio, he finds that the approach does not work well in real problems. He concludes that a more flexible channel model is needed. This seems reasonable, as many practical sources of nonlinear distortion, such as transducer overloading and tape saturation, are not memoryless.

6.1.3 Nonlinear autoregressive distortion

We allow for memory effects by modelling the distortion process using the NAR model considered in Chapter 5. The NAR model incorporates a nonlinear function which is approximated by a Volterra polynomial expansion of past sample values. As in Chapter 5, to avoid severe overfitting, it is necessary to select a subset of the many candidate polynomial terms.

6.1.3.1 Stepwise search

Mercer [141, Ch. 5,6] uses a similar model, but his approach is very different. He uses a stepwise regression procedure to choose model terms, with maximum likelihood parameter estimates. To reduce the computation required for each step of the algorithm, Mercer uses a coarse grid search followed by a local optimisation. After performing this selection procedure for several values of k, a model is chosen using the AIC (see §2.1.6.2).

As discussed in Chapter 5, the problem with stepwise regression and other deterministic search algorithms is that they can get stuck at local minima, *i.e.* subsets which are better than all the neighbouring ones but not the best overall. This is a particular problem in nonlinear model selection problems, as the search space tends to be highly multimodal.

6.1.3.2 Bayesian subset selection

We present here a fully Bayesian approach, implemented using MCMC methods, which has the advantage, in the case of model uncertainty, that model mixing can be used, so that the reconstruction is based on all the possible models, weighted according to their posterior probabilities, rather than just the single most probable one.
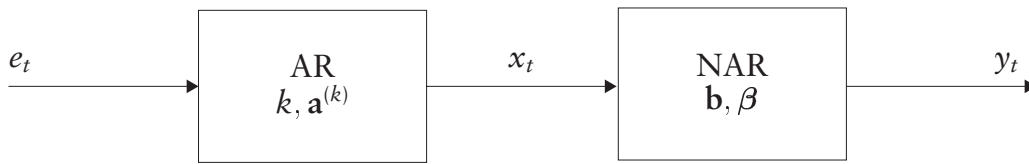
Figure 6.1. Cascade of AR source model and NAR distortion model.

6.2   *Model framework*

Figure 6.1 shows that we use a cascade model consisting of an AR source model, as used in Chapter 4, driving a NAR channel model, as used in Chapter 5.

6.2.1   Modelling equations

To recap, a linear AR model of order k can be expressed as (eq. 2.26)

$$x_t = e_t + \sum_{i=1}^{k} a_i^{(k)} x_{t-i} \qquad (6.1)$$

where

   $\{e_t\}$ is a zero-mean i.i.d. Gaussian excitation sequence
   $\{a_i^{(k)}\}$ are the parameters of the AR process.

and the triangularised polynomial NAR channel model can be written as (eq. 5.13)

$$y_t = x_t + \sum_{i=1}^{\eta_b} \sum_{j=1}^{i} \beta_{(i,j)} \, b_{(i,j)} \, y_{t-i} \, y_{t-j} + \sum_{i=1}^{\eta_b} \sum_{j=1}^{i} \sum_{k=1}^{j} \beta_{(i,j,k)} \, b_{(i,j,k)} \, y_{t-i} \, y_{t-j} \, y_{t-k}$$
$$+ \text{ higher degree terms} \quad (6.2)$$

where

   $\{y_t\}$ is the distorted signal we observe
   $\{x_t\}$ is the undistorted signal (eq. 6.1)

$\{b_{(i,j)}, b_{(i,j,k)}, \dots\}$ are the parameters of the NAR distortion process

$\{\beta_{(i,j)}, \beta_{(i,j,k)}, \dots\}$ are the corresponding binary indicators

η_b is the maximum lag of the NAR model

To avoid possible uniqueness problems when used in cascade with the linear AR model, purely linear terms have been excluded from the expansion [141]. As discussed in §6.5.1, this is not always necessary.

A major advantage of this model formulation is that the inverse of the nonlinear stage is a straightforward nonlinear moving average (NMA) filter, which is guaranteed to be stable. Hence it is simple to reconstruct the signal $\{x_t\}$ from $\{y_t\}$ for a given set of NAR parameters, $\{b_{(i,j)}, b_{(i,j,k)}, \dots\}$.

6.2.2   Subset & matrix-vector representation

Adopting the same matrix-vector notation as Chapters 4 & 5, we can express equations (6.1) & (6.2) as

$$\mathbf{e} = \mathbf{A}^{(k)}\mathbf{x} = \mathbf{x}_1 - \mathbf{X}^{(k)}\mathbf{a}^{(k)} \tag{6.3}$$

$$\mathbf{x} = \mathbf{y}_1 - \mathbf{Y}(\mathbf{b} \circ \boldsymbol{\beta}) \tag{6.4}$$

where

$\circ$ denotes the Hadamard (elementwise) product

$\mathbf{b}$ contains all the NAR parameters (eq. 5.14)

$\boldsymbol{\beta}$ contains the associated binary indicators

$\mathbf{x}_1$ omits the first k terms of $\mathbf{x}$

$\mathbf{y}_1$ omits the first η_b terms of $\mathbf{y}$

$\mathbf{A}^{(k)}$ and $\mathbf{X}^{(k)}$ are matrices containing elements from $\mathbf{a}^{(k)}$ and $\mathbf{x}$, respectively (eq. 2.31 & 2.30)

$\mathbf{Y}$ is a matrix containing products of elements of $\mathbf{y}$ (§5.4.2)

6.2.3   Likelihoods

The approximate likelihood for the linear model can be derived as (eq. 4.13)

$$p(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2) \approx p(\mathbf{x} \mid \mathbf{x}_0, k, \mathbf{a}^{(k)}, \sigma_e^2) \tag{6.5}$$

$$= \mathbf{N}\big(\mathbf{x}_1 - \mathbf{X}^{(k)}\mathbf{a}^{(k)} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{6.6}$$

where $\mathbf{x}_0$ contains the initial k elements of $\mathbf{x}$ and $\mathbf{I}_{n_e}$ is the identity matrix with the same number of rows as $\mathbf{e}$. Equation (6.6) can be rearranged to show that $\mathbf{x}$ is coloured zero-mean Gaussian noise which is approximately described by

$$\mathbf{x} \sim \mathbf{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{C_x}) \qquad \text{where} \qquad \mathbf{C_x}^{-1} = \frac{\mathbf{A}^{(k)T}\mathbf{A}^{(k)}}{\sigma_e^2} \tag{6.7}$$

and σ_e^2 is the variance of $\mathbf{e}$. The approximation is due to conditioning on $\mathbf{x}_0$, which is well known to be an insignificant end-effect unless n_e is small [23].

The approximate likelihood for $\mathbf{y}$ can hence also be expressed as a multivariate Gaussian (eq. 5.17):

$$p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_\beta, k, \mathbf{a}^{(k)}, \sigma_e^2) \approx p(\mathbf{y} \mid \mathbf{y}_0, \boldsymbol{\beta}, \mathbf{b}_\beta, k, \mathbf{a}^{(k)}, \sigma_e^2) \tag{6.8}$$

$$= \mathbf{N}\big(\mathbf{A}^{(k)}(\mathbf{y}_1 - \mathbf{Y}_\beta \mathbf{b}_\beta) \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \tag{6.9}$$

$$= \mathbf{N}\big(\mathbf{y}_1 - \mathbf{Y}_\beta \mathbf{b}_\beta \mid \mathbf{0}, \mathbf{C_x}\big) \tag{6.10}$$

where $\mathbf{y}_0$ is the first η_b elements of $\mathbf{y}$ and the notation $(\cdot)_\beta$ denotes a partition containing only elements corresponding to ones in $\boldsymbol{\beta}$, such that $\mathbf{Y}_\beta \mathbf{b}_\beta = \mathbf{Y}(\mathbf{b} \circ \boldsymbol{\beta})$.

6.2.4 Priors

We choose the same proper, conjugate, but fairly uninformative priors as used for the two separate parts of the model in §§4.3.2 & 5.4.4: bounded uniform for k, Bernoulli for $\boldsymbol{\beta}$, multiple independent Gaussians for $\mathbf{a}^{(k)}$ and $\mathbf{b}$, and inverse Gamma distributions for σ_a^2, σ_b^2 and σ_e^2.

6.2.5 Bayesian hierarchy

We wish to reconstruct the signal, $\mathbf{x}$. Doing this using equation (6.4) requires knowledge of $\boldsymbol{\beta}$ and $\mathbf{b}_\beta$, whose joint posterior is

$$p(\boldsymbol{\beta}, \mathbf{b}_\beta \mid \mathbf{y}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2) \propto p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_\beta, k, \mathbf{a}^{(k)}, \sigma_e^2)\, p(\boldsymbol{\beta})\, p(\mathbf{b}_\beta \mid \sigma_b^2)\, p(\sigma_b^2) \tag{6.11}$$

This, however, is dependent on k and $\mathbf{a}^{(k)}$, which are also unknown, and have posterior:

$$p(k, \mathbf{a}^{(k)} \mid \mathbf{x}, \sigma_a^2, \sigma_e^2) \propto p(\mathbf{x} \mid k, \mathbf{a}^{(k)}, \sigma_e^2)\, p(k)\, p(\mathbf{a}^{(k)} \mid \sigma_a^2)\, p(\sigma_a^2) \qquad (6.12)$$

which is dependent on $\mathbf{x}$.

6.3   *Markov chain Monte Carlo*

Since we cannot evaluate the required marginal distributions analytically, we again take the MCMC approach.

One of the great advantages of MCMC methods is that extra parameters and different types of move can easily be incorporated to tackle more complex problems. This results in a Markov chain with a mixture transition kernel [182], for which the same convergence results hold.

6.3.1   Reversible-jump moves for the linear stage

For this problem, we use, unchanged, the reversible-jump moves of §4.4.3 to accommodate uncertainty over k, the order of the linear AR model, together with the Gibbs sampling steps of §§4.5.1.1, 4.5.1.2 & 4.5.1.3 for the AR parameters, $\mathbf{a}^{(k)}$, and the hyperparameters.

6.3.2   Gibbs moves for the nonlinear stage

For subset selection and parameter estimation in the nonlinear model stage, the Gibbs sampling steps of §5.5.2.1 must be modified, as the NAR process is now excited by a non-white Gaussian signal, $\mathbf{x}$.

The derivation of the distribution from which to sample β_u is similar to that of §5.5.2.1, except that $\mathbf{C_x}^{-1} = \frac{\mathbf{A}^{(k)T}\mathbf{A}^{(k)}}{\sigma_e^2}$ (eq. 6.7) appears in the Gaussian terms, resulting in

$$p(\beta_u \mid \mathbf{y}, \beta_f, \mathbf{b}_{\beta_f}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2) \propto p(\beta_u) \sqrt{\frac{|\mathbf{C}_{cb_{\beta_u}}|}{|\mathbf{C}_{pb_{\beta_u}}|}} \exp\left(\tfrac{1}{2}\boldsymbol{\mu}_{cb_{\beta_u}}^T \mathbf{C}_{cb_{\beta_u}}^{-1} \boldsymbol{\mu}_{cb_{\beta_u}}\right)$$

$$(6.13)$$

Algorithm 6.1. Restoration of NAR distorted audio.

Choose initial values
for $i = \{1 \ldots$ number of iterations$\}$
    repeat
        Choose a group $(\cdot)_u$ of n_u nonlinear model terms to sample
        $\beta_u \sim p(\beta_u \mid \mathbf{y}, \beta_f, \mathbf{b}_{\beta_f}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2)$
        $\mathbf{b}_{\beta_u} \sim p(\mathbf{b}_{\beta_u} \mid \mathbf{y}, \beta, \mathbf{b}_{\beta_f}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2)$
    until all nonlinear model terms have been sampled
    $\mathbf{b}_\beta \sim p(\mathbf{b}_\beta \mid \mathbf{y}, \beta, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2)$
    $\sigma_b^2 \sim p(\sigma_b^2 \mid \mathbf{b}_\beta)$
    Reconstruct $\mathbf{x}$ from $\mathbf{y}$ using $\beta, \mathbf{b}_\beta$
    for $r = \{1 \ldots$ number of reversible-jump moves per iteration$\}$
        $k' \sim J(k' \mid k)$
        $z \sim U(0, 1)$
        if $z < \alpha(k \to k' \mid \mathbf{x}, \sigma_a^2, \sigma_e^2)$
            $k = k'$
        end if
        $\mathbf{a}^{(k)} \sim p(\mathbf{a}^{(k)} \mid \mathbf{x}, k, \sigma_a^2, \sigma_e^2)$
        $\sigma_a^2 \sim p(\sigma_a^2 \mid \mathbf{a}^{(k)})$
        $\sigma_e^2 \sim p(\sigma_e^2 \mid \mathbf{x}, k, \mathbf{a}^{(k)})$
    end for
 end for

---

where

$$C_{cb_{\beta_u}}^{-1} = Y_{\beta_u}^T C_{\mathbf{x}}^{-1} Y_{\beta_u} + C_{pb_{\beta_u}}^{-1} \qquad (6.14)$$

$$\boldsymbol{\mu}_{cb_{\beta_u}} = C_{cb_{\beta_u}} Y_{\beta_u}^T C_{\mathbf{x}}^{-1} (\mathbf{y}_1 - Y_{\beta_f} \mathbf{b}_{\beta_f}) \qquad (6.15)$$

and Y_{β_u} contains those columns of Y which correspond to ones in β_u.

Again, $\mathbf{b}_{\beta_u}$ is drawn in a simple Gibbs move from a multivariate Gaussian:

$$p(\mathbf{b}_{\beta_u} \mid \mathbf{y}, \beta, \mathbf{b}_{\beta_f}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2) \propto \mathbf{N}\big(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{cb_{\beta_u}}, C_{cb_{\beta_u}}\big) \qquad (6.16)$$

and all of $\mathbf{b}_\beta$ can be sampled jointly in a similar move (§5.5.2.3). The Gibbs sampling step for the hyperparameter, σ_b^2, is unchanged.

6.3.3   Sampling strategy

Since **x** is relatively expensive to compute, particularly when processing long signals (see §6.5.1), we separate our sampling moves into those which affect the linear AR model and those which affect the nonlinear model. In each iteration, these two groups are sampled in turn—see Algorithm 6.1. Sensible initial values for the parameters could be drawn from their prior distributions. As we are using quite uninformative priors, we choose to start with null models, *i.e.* $k = 0$ and $\beta = 0$.

6.4   Experiments with single blocks

6.4.1   Synthetic data

To test the model estimation procedure, 8000 samples were generated from a synthetic AR-NAR process with AR order 6 and six nonlinear terms:

$$
\begin{aligned}
b_{(2,2)} &= -0.20 & b_{(4,1)} &= 0.18 & b_{(5,4,1)} &= -0.16 \\
b_{(5,5,3)} &= 0.16 & b_{(5,5,5)} &= 0.20 & b_{(2,2,2,1)} &= -0.10
\end{aligned}
\tag{6.17}
$$

Figure 6.2 shows the result of running the sampler for 2000 iterations with 82 candidate nonlinear terms (second and third degree to lag 6 and fourth degree to lag 2). Indicators were sampled in random triples, and eight reversible-jump moves were proposed each iteration. It was initialised with an empty model and arbitrary values for σ_e^2, σ_a^2 and σ_b^2. Because the initial values are atypical of the posterior distribution, as is usually the case with problems in high dimensions, the beginning of the run was discarded as burn-in and only values from the final 1000 iterations were used for analysis.

Figure 6.2d–e shows that the sampler converged very quickly to the correct AR model order. The six nonlinear model terms which appear most frequently in the sampler output (Figure 6.2a–b) are correct; that subset accounts for over 50% of the iterations (Figure 6.2c).

Figure 6.3 shows Monte Carlo estimates of the posterior distributions of the parameter values, produced from those iterations which selected the most popular model. It can be seen that the estimated distributions have
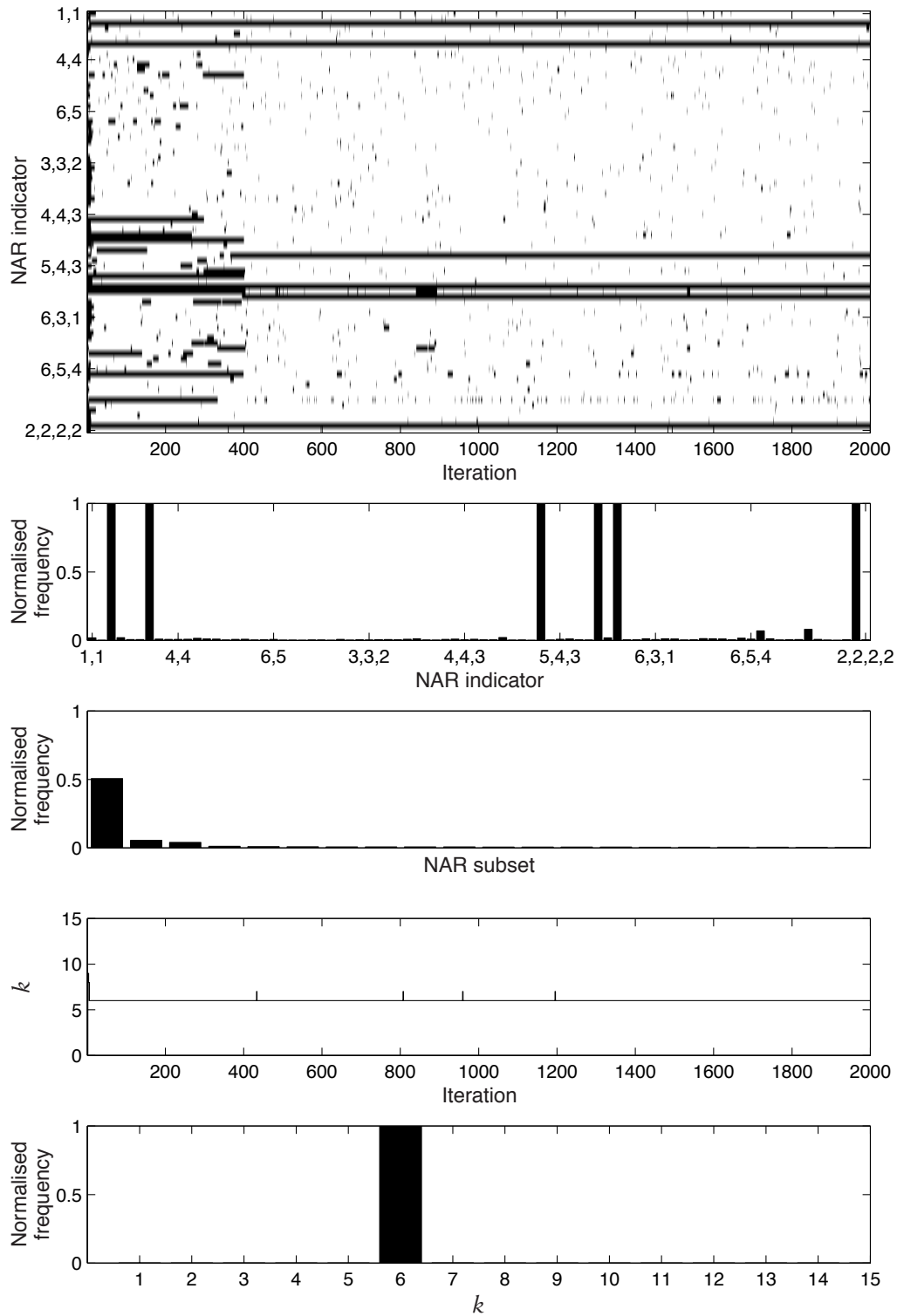
Figure 6.2. Identifying synthetic AR-NAR data: (from top) (a) *raw β values—black areas represent ones;* (b) *frequency of appearance of each NAR model term;* (c) *frequency of appearance of the most popular NAR subsets;* (d) *raw k values;* (e) *frequency of appearance of each k value.*
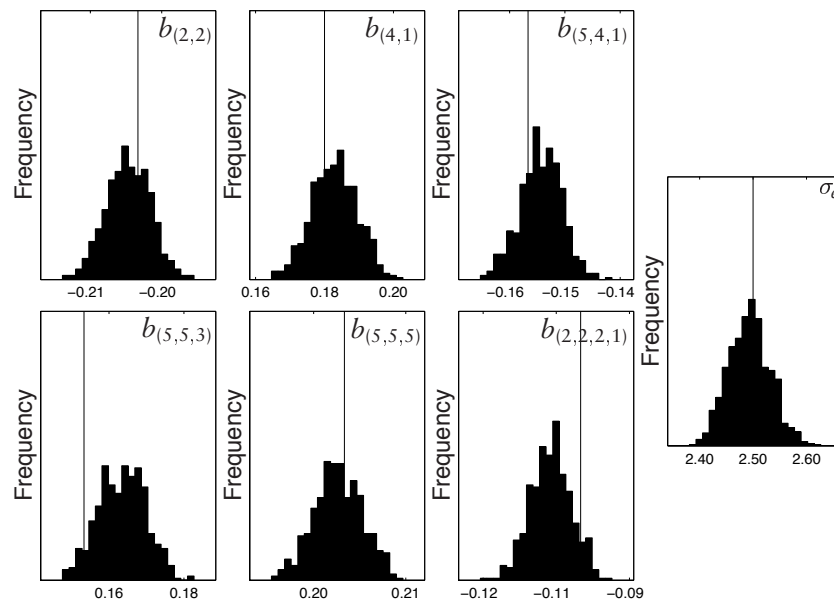
Figure 6.3. Identifying synthetic AR-NAR data: *histograms of σ_e and NAR parameter values (true values marked by lines).*
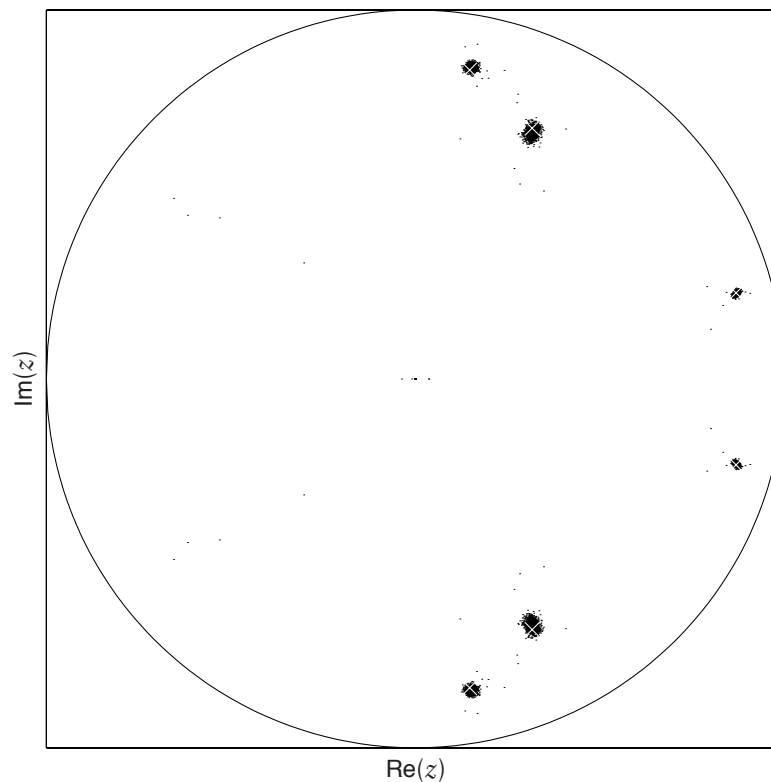
Figure 6.4. Identifying synthetic AR-NAR data: *sampled linear AR pole positions (true values marked by white crosses).*
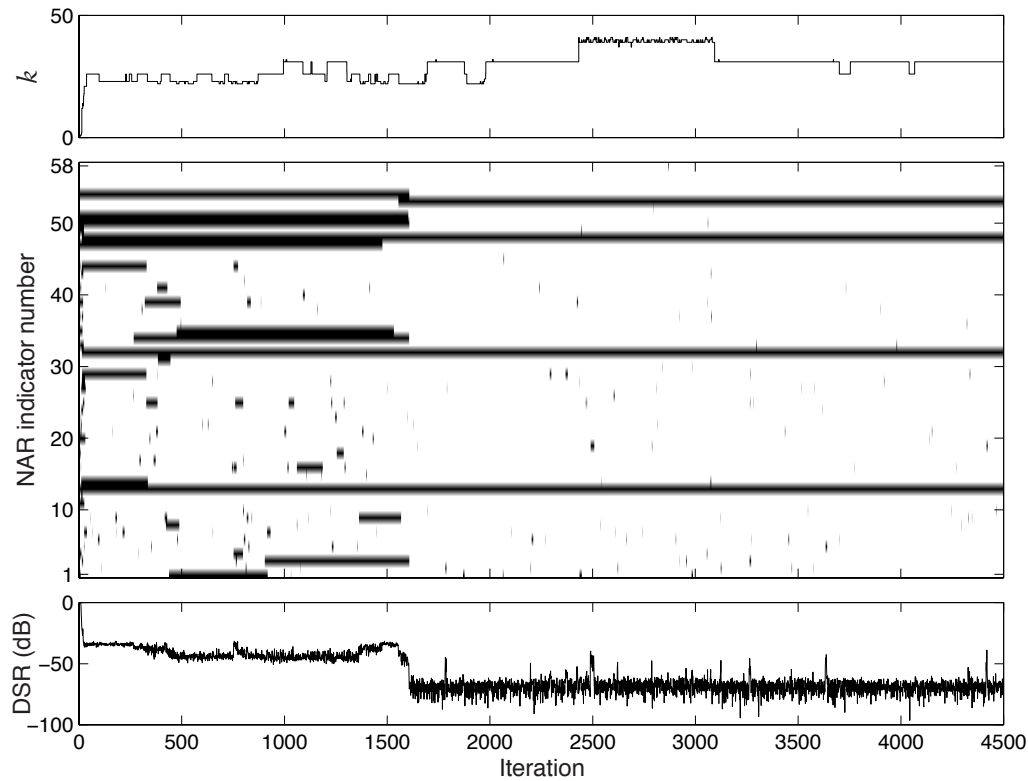
Figure 6.5. Synthetically distorted pop music extract: *(from top): (a) AR model order, k; (b) NAR indicators, β (black pixels indicate $\beta_i = 1$); (c) distortion-to-signal ratio for the reconstructed signal.*

substantial probability mass close to the known true values. The scatter plot in Figure 6.4 shows that the AR parameters were also accurately estimated.

6.4.2   Synthetically distorted audio

1000 samples from a 44.1 kHz pop music recording were distorted by the following NAR filter:

$$y_t = x_t - 0.15y_{t-2}^2\,y_{t-1} + 0.2y_{t-2}\,y_{t-1}^3 - 0.08y_{t-2}^2\,y_{t-1}^3 + 0.005y_{t-2}\,y_{t-1}^4 \tag{6.18}$$

This introduced distortion 11 dB below the r.m.s. signal level. We call this a distortion-to-signal ratio (DSR) of -11 dB.

The sampling scheme was run for 4500 iterations, again starting with empty models and arbitrary initial hyperparameter values, with 58
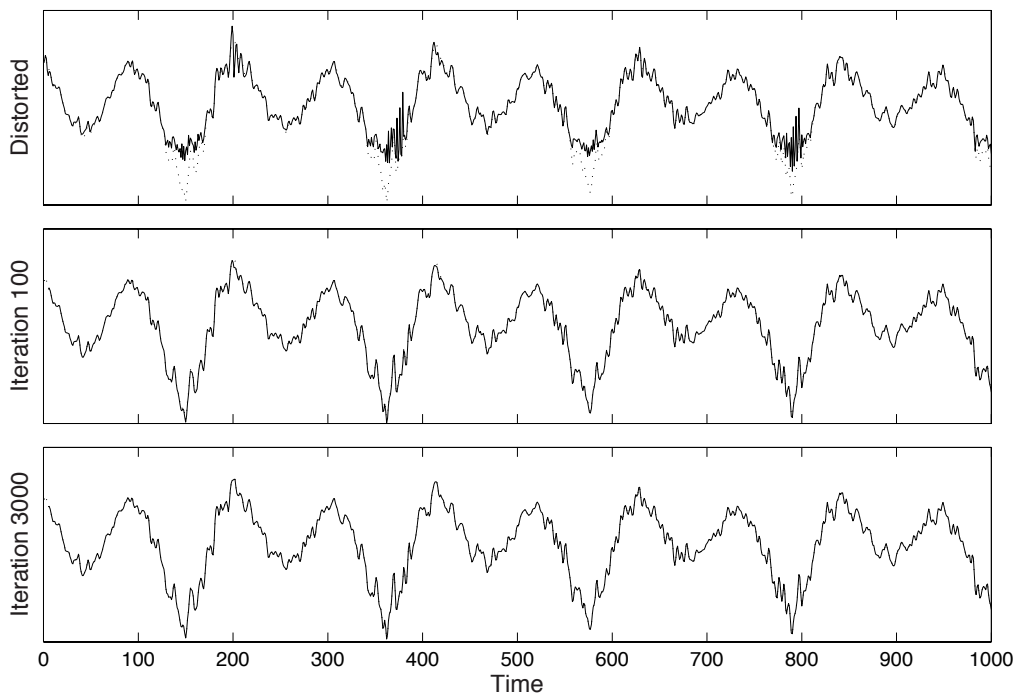
Figure 6.6. Comparison of restorations of pop music extract: *original signal* (dotted) *compared with* (solid) *distorted signal, restoration after 100 iterations and restoration after 3000 iterations.*

candidate nonlinear terms (second and third degree up to lag 4, fourth degree up to lag 3 and fifth and sixth degree up to lag 2), and allowing AR models up to order 100. Figure 6.5 shows the sampler output together with the DSR. It can be seen that after 1700 iterations the correct nonlinear terms have been chosen and the reconstruction is almost perfect—the DSR is improved to an average of around -69 dB. It is interesting that, even after as few as 100 iterations, the DSR is consistently around -34 dB, a useful improvement of 23 dB. Figure 6.6 shows the initial, distorted waveform, the reconstruction after 100 iterations and the reconstruction after 3000 iterations, each with the original, undistorted waveform for comparison.

6.5 Extension to long signals

We now consider how to apply this method to the restoration of distorted audio signals. A conventional approach to speech and audio modelling is to break the signal into blocks which are sufficiently short that it is

reasonable to assume stationarity (see *e.g.* [89]). Typical block lengths are around 25 ms.

6.5.1   Joint estimation over multiple blocks

A crude approach would be to process each block separately. With many distortion problems, however, we can expect the distortion process to remain unchanged for the duration of the signal, which could be many minutes. This can be exploited by estimating $\mathbf{b}$ and β over many (not necessarily contiguous) blocks of audio, $\mathbf{y}_{[r]}$, $r \in \{1 \ldots R\}$, the source model for each of which has separate parameters $k_{[r]}$, $\mathbf{a}_{[r]}^{(k)}$ and $\sigma_{e\,[r]}^2$.

This changes the structure of $\mathbf{C_x}^{-1}$ (eq. 6.7), which is required when sampling the global parameters β and $\mathbf{b}$. It can, however, be computed quickly, since

$$\mathbf{C_x}^{-1} = \sum_{r=1}^{R} \frac{1}{\sigma_{e\,[r]}^2} \tilde{\mathbf{A}}_{[r]}^{(k_{[r]})T} \tilde{\mathbf{A}}_{[r]}^{(k_{[r]})} \tag{6.19}$$

where $\tilde{\mathbf{A}}_{[r]}^{(k_{[r]})}$ is $\mathbf{A}_{[r]}^{(k_{[r]})}$ padded left and right with an appropriate number of columns of zeros. The computation required hence increases only linearly with the number of blocks, and much can be performed in parallel.

This multiple block technique is a powerful approach, as it allows large quantities of data to be processed at once, all contributing to the model selection process. As long as the signal does vary significantly over its duration, linear model terms can be permitted in the channel model without introducing ambiguity. This introduces the possibility of modelling linear phenomena such as reverberation as well as purely nonlinear effects.

6.5.2   Long audio signal

A four second extract from a 44.1 kHz sampled orchestral recording (Track 6 on the accompanying CD—see Appendix C) was artificially distorted by a NAR filter with the following parameters:

$$b_{(1,4,6)} = -0.07 \quad b_{(2,2,3)} = -0.05 \quad b_{(3,6,8)} = -0.06$$
$$b_{(4,7,7)} = -0.06 \quad b_{(8,9,9)} = -0.05 \tag{6.20}$$
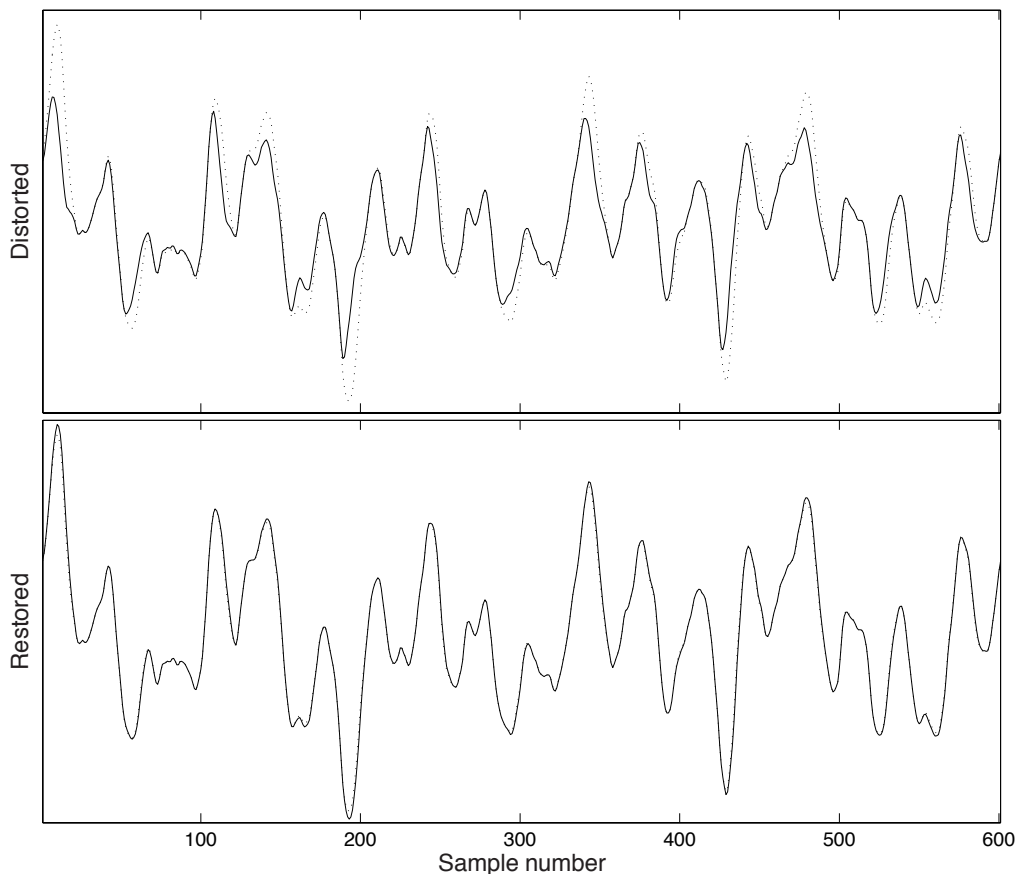
Figure 6.7. Restoring a long audio signal: *part of the original signal* (dotted) *with* (solid, from top)*: distorted signal* **y***; restored signal,* **x̂**.

The resulting signal has a DSR of -10.5 dB, and the degradation is clearly audible (Track 7).

Thirty randomly chosen, non-contiguous blocks of 1000 samples were used for analysis, and the sampler was run for 500 iterations with all 220 third degree nonlinear terms up to lag 10 as candidates. The model space is so large that no subset appears more than once in the sampler output. However, from the 50th iteration onwards, the DSR of the restoration is always better than -23.6 dB. Allowing model mixing by making a Monte Carlo estimate of **x** directly from the final 250 iterations decreases (*i.e.* improves) the DSR to -27.8 dB. Since the model is linear-in-the-parameters [151], this is equivalent to performing a restoration using a single NMA filter incorporating all the terms which appear in the sampler output, with the parameters averaged over all the iterations, treating excluded terms' parameters as zeros.

Figure 6.7 compares the waveforms of part of the original signal with the distorted signal and the model-averaged restoration. It can be seen that the restored signal matches the original signal closely. The restored signal (Track 8) is hard to distinguish from the original (Track 6).

6.6   Discussion

6.6.1   Real audio distortion

Physical distortion-producing processes, such as tape saturation, are often followed by linear processes, such are playback equalisation. Extremely large Volterra expansions can be required to model these adequately, so future work will consider modelling these explicitly as a third, either AR or MA, stage in the cascade model. The parameters of this extra stage can then be estimated jointly across the whole duration of the signal. The effects of bandlimiting could be similarly incorporated [154]. It may also be advantageous to model explicitly any background noise or outliers caused by defects in the recording medium. The latter can be done using further indicator variables within the same MCMC scheme [84–86, 88].

6.6.2   Conclusions

The novel MCMC method presented here jointly estimates the structure and parameters of a cascade AR-NAR model. Using the efficient reversible-jump proposal distributions and joint Gibbs sampler moves developed in previous chapters, we have exploited the partially analytic structure of both the linear and nonlinear parts of the model to speed the convergence of the Markov chain.

This approach allows for model mixing, which is important in this application as there is uncertainty: often no single nonlinear model dominates the posterior. It also allows estimation of a fixed channel model from observation of a long, time-varying signal without a dramatic increase in computation.

Quantisation distortion

<div style="text-align: right">7</div>

7.1   Quantisation problem

For digital processing, transmission or storage, a signal is represented as discrete in both time (due to sampling) and value (due to quantisation). Nyquist [145] showed that, in the absence of quantisation, the sampling process is lossless if the signal is bandlimited to below half the sampling frequency.

The quantisation process, however, introduces an error component, often referred to as *quantisation noise*. Since this quantisation error is signal-dependent (as we shall see in §7.1.3), it is perhaps better described as *quantisation distortion*.

7.1.1   Word length in digital audio

In digital audio systems, the quantised sample values are usually stored as binary numbers representing signed integers. It is normal to describe the quantisation used in a digital system by the number of bits used to represent each sample. This is called *word length* or *bit depth*.

Table 7.1 shows the maximum and minimum integer values which can be stored in words of various lengths when using two's-complement representation. It also shows the approximate r.m.s. power of the quantisation error. This is expressed in dBFS, where 0 dBFS is defined as the r.m.s. power of a full scale sine wave [1]. It can be seen that, as expected, each additional bit halves the quantisation step height, and hence increases the *dynamic range* by about 6 dB.

Table 7.1. Typical word lengths used in audio.

Word length (bits)	Integer range	Quantisation level (dBFS, r.m.s.)
8	-128 – 127	-48
12	-2048 – 2047	-72
16	-32768 – 32767	-96
20	-524288 – 524287	-120
24	-8388608 – 8388607	-144
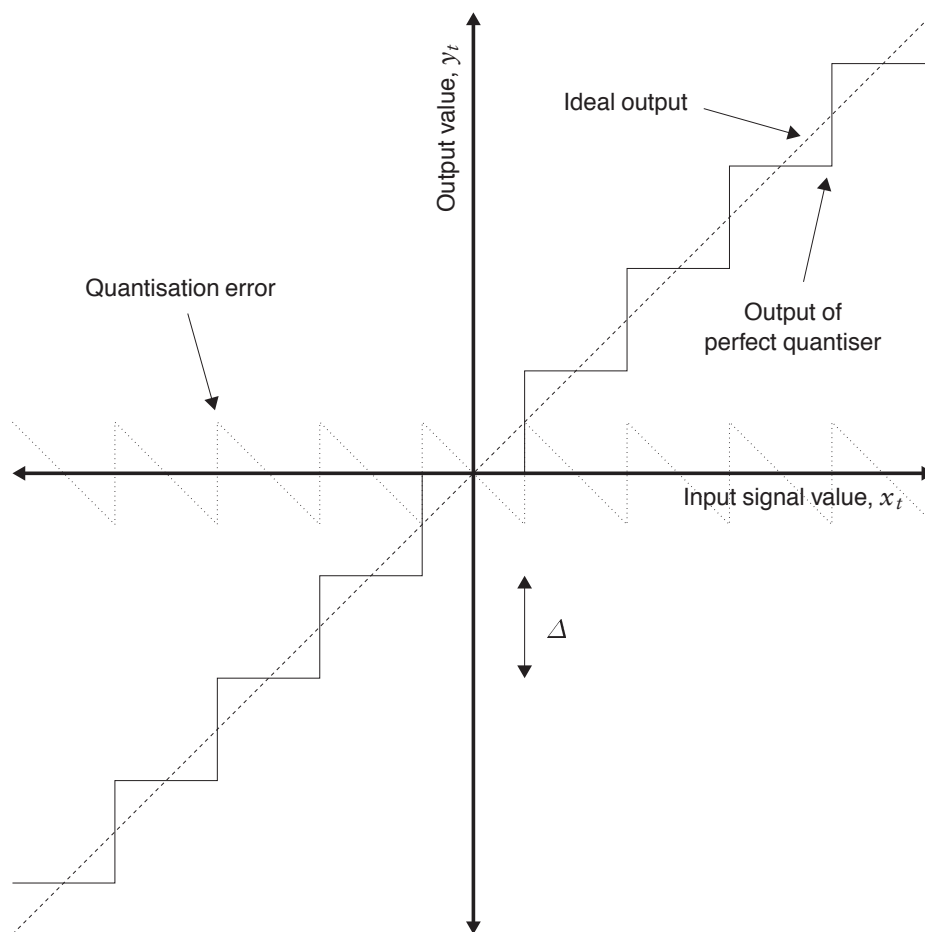
Figure 7.1. Transfer function of, and quantisation error introduced by, a perfect mid-tread quantiser with step height Δ.

7.1.2 Perfect quantiser

The quantisation process is memoryless and deterministic:

$$y_t = Q(x_t) \tag{7.1}$$

where $\{x_t\}$ is the original signal, $\{y_t\}$ is the quantised signal, and $Q(\cdot)$ is the quantisation function. Figure 7.1 shows the transfer function of a perfect quantiser. If the $\Delta = 1$, then $Q(\cdot)$ is equivalent to rounding to the nearest integer. In practical analogue-to-digital converters, quantisation is often far from perfect, possibly exhibiting uneven step heights and missed codes (see *e.g.* [196]). As long as these defects are memoryless, they can be incorporated straightforwardly into $Q(\cdot)$.

Quantisation is also performed in the digital domain: when signal values are involved in multiplication, in order to adjust the level of the signal or perform equalisation, the results are generally non-integer. The arithmetic operations are usually performed at high precision[1], but the result must be requantised before it can be stored in memory of limited word length. The cheapest way to do this is simply to truncate the binary representation of the signal at the number of bits required. This is essentially equivalent to rounding and adding a fixed offset.

7.1.3 Quantisation distortion

Figure 7.2 shows the effect of quantising a very low amplitude (-84 dBFS on a 16 bit system) sine wave. The resulting error, of peak-to-peak amplitude Δ, is clearly signal dependent.

Figure 7.3a shows the corresponding periodogram estimates of the power spectrum. The quantisation distortion looks like odd harmonic distortion and a fairly flat noise floor, which one would expect to sound quite innocuous. In fact, using a longer data window to allow us to resolve finer detail (fig. 7.3b) shows that the distortion consists entirely of isolated sinusoids: quantisation has introduced harmonics; those above the Nyquist frequency (22.05 kHz) have repeatedly been aliased back into the baseband, where they are inharmonic.

The structure of the distortion is particularly clear here because the

---

[1]Many microprocessors used for audio perform fixed point arithmetic with word lengths of at least 24 bits; floating point processors typically maintain a mantissa of length 24 or 52 bits.
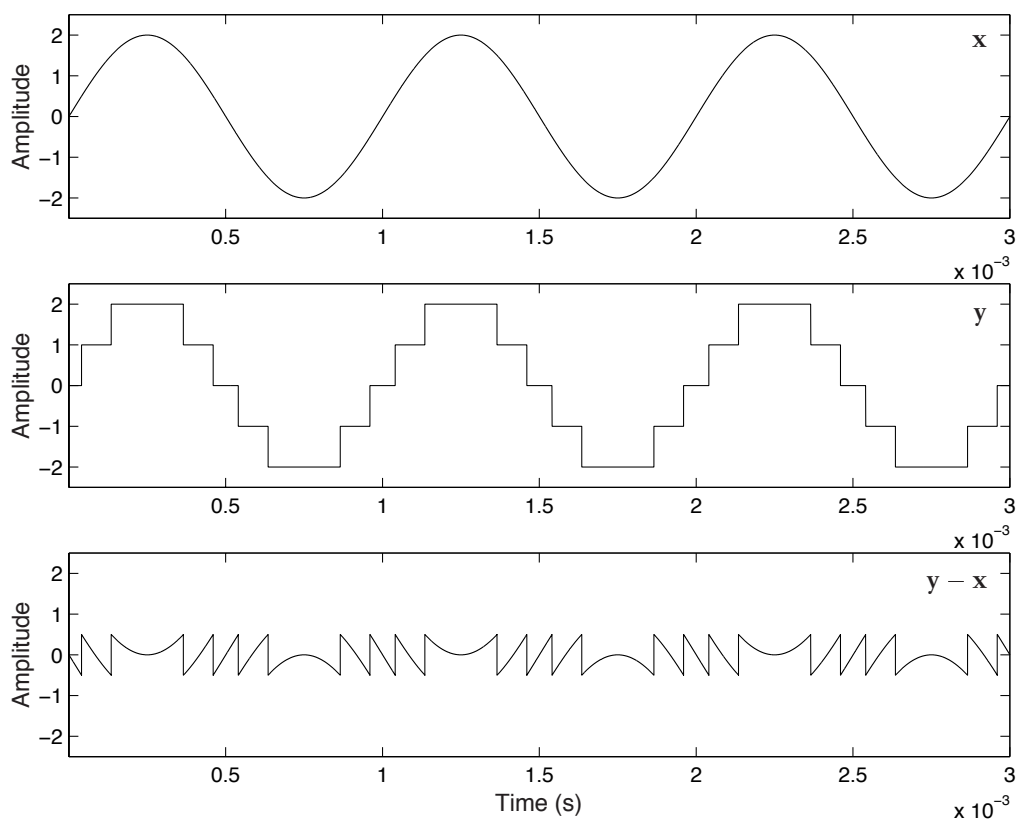
Figure 7.2. Illustration of quantisation distortion: (top) *a 1 kHz sine wave of amplitude 2,* (middle) *the same signal, quantised with a step height of 1,* (bottom) *quantisation error signal. (After [127].)*

components are at intervals of 100 Hz (the highest common divisor of 1000 and 44 100). In general, they will be much more closely spaced; for complex signals the spectrum of the quantisation distortion becomes almost white, making it much less noticeable.

Two features of the distortion spectrum are particularly disturbing in audio:

Aliasing When the fundamental frequency of a note varies slightly, the harmonics vary with it, as expected, but those which have been aliased will vary in the opposite direction.[2]

Level sensitivity The relative levels of the various harmonics and aliased harmonics can change significantly when the input signal amplitude changes slightly, giving rise to *granulation noise* [135].

This second phenomenon is often heard on decaying musical notes, where

---

[2] . . . unless they have been aliased twice, or any other even number of times.
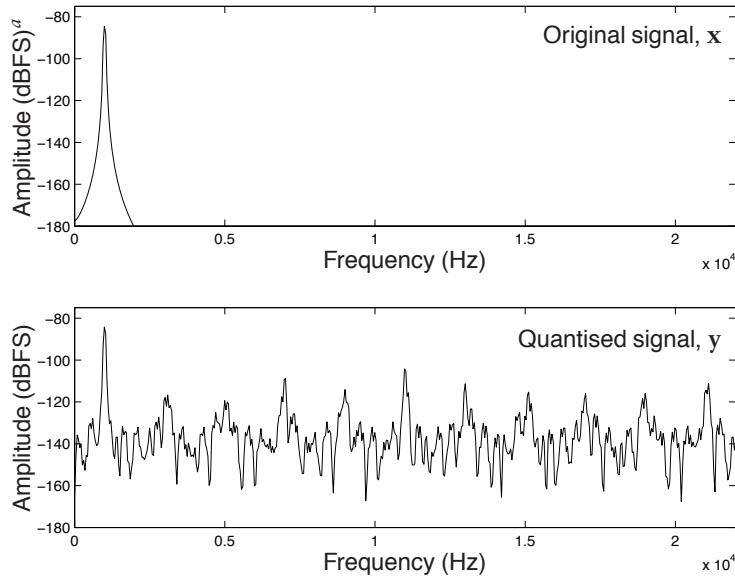
(a) *Complete periodogram (1024 bins)*
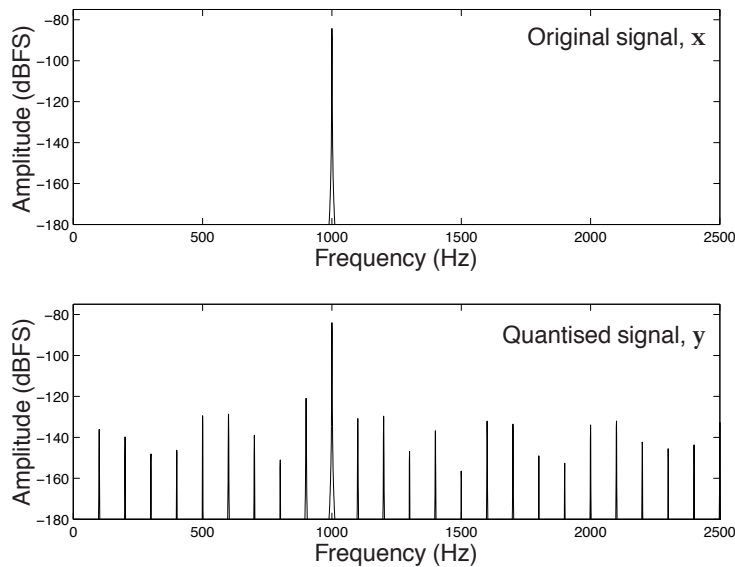
(b) *Detail from high resolution periodogram (2^{16} bins)*

Figure 7.3. Estimated power spectra of quantised signal: *These spectra correspond to the 1 kHz sine wave of Figure 7.2 before and after 16 bit quantisation, assuming a sample rate of 44.1 kHz. Power spectrum estimates were produced using the FFT on overlapping Hanning-windowed blocks.*

[a]The amplitude scale is such that a full scale sine wave on a 16 bit system would appear as a 0 dBFS peak, in accordance with [1].
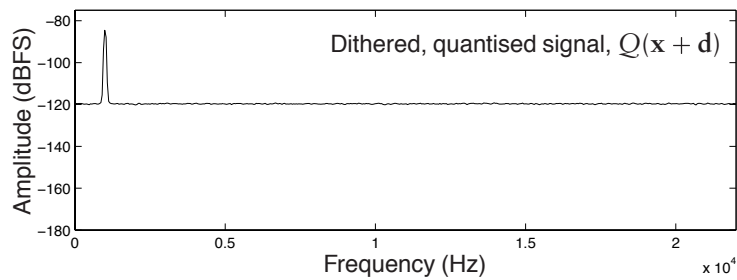
Figure 7.4. Estimated power spectrum after dithered quantisation: *The 1 kHz sine wave of Figure 7.2 was quantised after adding TPDF dither,* **d***, with range* ± 1*. Contrast with Figure 7.3a.*

it is exacerbated by the tendency of many instruments to produce a more sinusoidal waveform (with a correspondingly peakier distortion spectrum) as a note decays. Decaying piano notes are a good example of this. Track 9 on the accompanying CD (see Appendix C) is finely quantised ($\Delta = 1$), such that the distortion is not noticeable, whereas Track 10 is quantised with $\Delta = 50$. This introduces distortion 23 dB below the r.m.s. level of the signal, where it is clearly audible. Track 11 contains just the difference signal, *i.e.* the added distortion, at a higher level. The change in character, from apparently random noise while the notes are at high amplitude to obviously signal-correlated distortion as they decay, can be clearly heard.

7.1.4   Dither

There has been much research into the effects of quantisation and their elimination at the time of quantisation through the use of dither (see Lipshitz *et al.* [127] for a comprehensive survey), some of which is reviewed below.

7.1.4.1   Additive dither

Dithering, a technique originally developed to avoid banding in quantised video images [164], involves adding a low-level random component to the signal before quantisation. If the probability distribution of the dither signal meets certain criteria [127], certain moments of the quantisation error will become uncorrelated with the signal.

There is a wide range of dither probability distributions which can be used, but for audio the optimum choice is a triangular p.d.f. (TPDF) with range $\pm \Delta$, as this is the lowest power dither which decorrelates the first and

second moments of the quantisation error from the signal, which is all that is required for it to be perceived as white noise.

Figure 7.4 shows the effect of TPDF dither on the spectrum of the quantisation error of the 1 kHz sine wave considered in §7.1.3. It can be seen that the distortion components have vanished. The tradeoff for this improvement is that the noise floor is now 4.7 dB higher.

The dither makes the *least significant bit* of the quantised signal keep changing state such that

$$p\big(Q(x_t + d_t) = Q(x_t) + \Delta\big) \propto \frac{x_t - Q(x_t)}{\Delta} \tag{7.2}$$

where $\{d_t\}$ is the dither component. Hence the expected value of the least significant bit is equal to the fractional value needed to accurately represent the signal. This can be thought of as a pulse probability modulation scheme. The effect is to linearise the time-averaged transfer function.

As an extreme example, if the entire signal $\mathbf{x}$ lies within one step of the quantiser, quantisation without dither gives silence, whereas quantisation with TPDF dither results in a signal with one active bit. Since human hearing integrates over time, the signal can still be perceived, albeit below the noise floor.

7.1.4.2   Noise shaping

Noise shaping [202] controls the spectral shape of the raised noise floor by introducing a feedback loop around a dithered quantiser. By careful choice of a filter in the feedback path, most of the dither energy can be put in frequency bands at which the ear is less sensitive, or where the noise will be attenuated by the reconstruction filter in a digital-to-analogue converter. Hence, although the r.m.s. noise level necessarily increases, the perceived noise level can be lower. See Gerzon & Craven [69] for a clever application.

7.1.4.3   Subtractive dither

A better way to avoid problems with dither noise is to subtract the dither signal after quantisation [127]. Unfortunately, the need to recreate the dither signal with correct synchronisation at the time of conversion back to analogue (or to a longer word length) has so far limited its use to niche areas (see *e.g.* [49]).
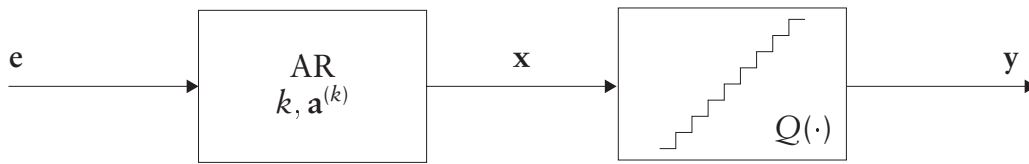
Figure 7.5. Modelling of quantised audio: **e** *is the excitation process,* **x** *the undistorted audio signal, and* **y** *the observed, quantised audio signal.*

7.1.5   The restoration problem

Despite the simplicity with which quantisation distortion can be avoided using dither, much current audio equipment and software does not implement it properly or at all. There is therefore a large catalogue of recordings exhibiting quantisation distortion. There does not, however, appear to be any published work on reconstructing a signal when only a coarsely quantised version is available. It is this problem which we seek to address.

Of course, if some of the information content of the signal has been lost in the quantisation process, such as parts which are entirely below the quantisation level, it cannot be recovered.

It seems likely, however, that what is most objectionable about quantisation distortion is the addition of inharmonic components, rather than any information loss. If this is so, it should be possible to achieve an audible improvement.

7.2   *Restoration using an AR signal model*

As shown in Figure 7.5, we model the signal and the quantisation process explicitly as a cascade model (§2.3.4). Although this approach is similar in concept to the AR-NAR cascade restoration of Chapter 6 and the AR-MNL method (§6.1.2.3, [141]), in this case we assume that we know $Q(\cdot)$; the problem is that it does not have a unique inverse. Whereas before we were using the linear signal model to enable model selection and parameter estimation for the nonlinear stage, from which **x** followed directly, now we use the linear model to estimate **x**.

Initially, we use an AR model for the signal, with unknown order k, parameters $\mathbf{a}^{(k)}$ and i.i.d. Gaussian excitation with variance σ_e^2. As in §4.7 and Chapter 6, the time-varying audio signal is broken up into a series of

blocks, each with separate parameters k, $\mathbf{a}^{(k)}$ and σ_a^2. To avoid confusing notation, the following derivations refer only to one block's parameters.

Taking the Bayesian approach, inference concerning $\mathbf{x}$ is made on the basis of the marginal posterior distribution $p(\mathbf{x} \mid \mathbf{y})$. Since this cannot be evaluated directly, the joint posterior $p(\mathbf{x}, k, \mathbf{a}^{(k)}, \sigma_e^2 \mid \mathbf{y})$ will be simulated and a Monte Carlo estimate made of $\mathbf{x}$.

7.2.1 Sampling k, $\mathbf{a}^{(k)}$ and σ_e^2

The model order selection and parameter estimation can be performed in a reversible-jump MCMC framework, using the priors and methods developed in Chapter 4, conditioning on the current restoration, $\mathbf{x}$, in the same manner as in Chapter 6.

7.2.2 Sampling $\mathbf{x}$

The quantisation process is a many-to-one mapping, so its inverse is one-to-many and provides the range of possible values that the input might have taken, *i.e.*

$$x_t \in Q^{-1}(y_t) \tag{7.3}$$

In terms of probability distributions,

$$p(\mathbf{y} \mid \mathbf{x}, \ldots) = \delta\left(\mathbf{y} - \mathbf{Q}(\mathbf{x})\right) \tag{7.4}$$

where $\delta(\cdot)$ is a Dirac delta function and $\mathbf{Q}(\cdot)$ is a vector form of $Q(\cdot)$. Applying Bayes' theorem (eq. 2.3) gives

$$p(\mathbf{x}_1 \mid \mathbf{y}, k, \mathbf{a}^{(k)}, \sigma_e^2, \mathbf{x}_0) \propto \begin{cases} p(\mathbf{x}_1 \mid k, \mathbf{a}^{(k)}, \sigma_e^2, \mathbf{x}_0) & \text{if } \mathbf{x} \in \mathbf{Q}^{-1}(\mathbf{y}) \\ 0 & \text{elsewhere} \end{cases} \tag{7.5}$$

where $p(\mathbf{x}_1 \mid k, \mathbf{a}^{(k)}, \sigma_e^2, \mathbf{x}_0)$ is the conditional likelihood for an AR model (eq. 2.34).

Since it is not computationally feasible to sample from the whole of $\mathbf{x}$ directly (see §7.2.3), a Gibbs sampler (§3.3.1.3) approach is taken, in which a subblock, $\mathbf{x}_u$, is updated conditional on the rest of the block, $\mathbf{x}_f$, which remains fixed. This is repeated, with different partitioning, until all subblocks

(*i.e.* the whole block) have been sampled.

The AR model (eq. 2.27) can be partitioned to update one subblock as follows:

$$\mathbf{e} = \mathbf{A}^{(k)}\mathbf{x} = \mathbf{A}_u^{(k)}\mathbf{x}_u + \mathbf{A}_f^{(k)}\mathbf{x}_f \tag{7.6}$$

where $(\cdot)_f$ also contains $\mathbf{x}_0$ and the first k samples from the following block, to ensure continuity across the block boundaries.[3] $(\cdot)_u$ need not be contiguous, but since it is likely that adjacent samples will be more highly correlated, it may be advantageous to sample them jointly (§3.5.3.2).

The full conditional distribution for $\mathbf{x}_u$ is the same as that required for interpolation (see *e.g.* [87, 146]), except that now it is bounded to lie within $\mathbf{Q}^{-1}(\mathbf{y}_u)$. If the bounds are represented by the function

$$B(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathbf{Q}^{-1}(\mathbf{y}) \\ 0 & \text{elsewhere} \end{cases} \tag{7.7}$$

then equation (7.5) can be rearranged as

$$p(\mathbf{x}_u \,|\, \mathbf{y}, \mathbf{x}_f, k, \mathbf{a}^{(k)}, \sigma_e^2) \tag{7.8}$$

$$= B(\mathbf{x}_u)\, p_{\mathbf{e}}(\mathbf{A}_u\mathbf{x}_u + \mathbf{A}_f\mathbf{x}_f) \tag{7.9}$$

$$= B(\mathbf{x}_u)\, \mathbf{N}(\mathbf{A}_u\mathbf{x}_u \,|\, -\mathbf{A}_f\mathbf{x}_f, \sigma_e^2) \tag{7.10}$$

which can be rearranged using equation (A.5) as

$$\propto B(\mathbf{x}_u)\, \mathbf{N}(\mathbf{x}_u \,|\, -(\mathbf{A}_u^T\mathbf{A}_u)^{-1}\mathbf{A}_u^T\mathbf{A}_f\mathbf{x}_f, \sigma_e^2(\mathbf{A}_u^T\mathbf{A}_u)^{-1}) \tag{7.11}$$

which is a multivariate Gaussian distribution, bounded to a hypercube[4] in $\mathbf{x}_u$-space.

7.2.3 Sampling bounded Gaussians

The *n*-dimensional multivariate Gaussian distribution,

$$\mathbf{N}(\boldsymbol{\theta} \,|\, \boldsymbol{\mu}, \mathbf{C}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{C}|^{\frac{1}{2}}} \exp\left(-\tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})\right) \tag{7.12}$$

---

[3]In fact the structure of $(\mathbf{A}_u^T\mathbf{A}_u)^{-1}\mathbf{A}_u^T\mathbf{A}_f$ (see eq. 7.11) is such that the $(\cdot)_f$ partition need only contain the nearest k samples to each side of the $(\cdot)_u$ partition.

[4]This is for a perfect quantiser; more generally it will be a hypercuboid region.

is very widely used. Since it can be expressed as a linear transformation of multiple independent univariate Gaussians (eq. A.5), samples can be drawn from it by drawing a vector ϕ of n independent samples from a zero-mean, unit variance Gaussian distribution, then applying the transformation

$$\theta = \mathbf{U}^T \phi + \mu \tag{7.13}$$

where the matrix $\mathbf{U}$ such that $\mathbf{U}^T\mathbf{U} = \mathbf{C}$ is found by a matrix square-root method such as the Singular Value Decomposition or the Cholesky decomposition (see *e.g.* [62, p. 478]). The univariate Gaussians can be sampled very efficiently by transformation of rectangularly distributed samples using the Box-Muller method (see *e.g.* [155]).

When bounds are introduced, however, it becomes surprisingly difficult to draw samples efficiently. We now look at how some of the sampling methods discussed in §3.4 can be applied to the problem.

7.2.3.1 Univariate bounded Gaussians

Rejection sampling   The most obvious method for producing samples from a bounded Gaussian distribution is to draw samples from the full distribution, and reject those which are outside the bounds. The expected value of the acceptance rate will equal the proportion of the distribution which lies within the bounds:[5]

$$E(\alpha) = \int_{b_{\min}}^{b_{\max}} \mathrm{N}(\theta \mid 0, 1)\, d\theta \tag{7.14}$$

Clearly this will become unacceptably small if the range $(b_{\min}, b_{\max})$ lies out in the tails of the distribution. In this case α can be doubled simply by exploiting the symmetry of the Gaussian distribution to remap samples from the other side of the mean which would otherwise be rejected.

For the case where the range is $(b_{\min}, \infty)$ and $b_{\min} > 0$ (or the equivalent range in the negative tail of the distribution), Marsaglia [138] generates proposals from the tail of a Rayleigh distribution, then uses rejection sampling (§3.4.2) to correct it to the required Gaussian. The expected acceptance rate exceeds that of equation (7.14) (with remapping) for $b_{\min} > 0.65$, and approaches 1 as $b_{\min} \to \infty$.

---

[5]Without loss of generality, we consider only the zero-mean Gaussian distribution with unit variance; simple transformations will give distributions with other means and variances.

Robert [162] improves on this method by using an exponential proposal distribution. He then considers the case where $b_{\max} < \infty$, providing two rejection sampling algorithms, suitable for different ranges of $b_{\min}$ and $b_{\max}$, such that a reasonably high expected acceptance rate is obtained for any combination. Geweke [70] presents another variation, using a combination of uniform, exponential, normal and half-normal rejection sampling.

Inverse cumulative distribution function   A conceptually simpler approach is that of §3.4.1: to use the inverse of the c.d.f., $\Phi^{-1}(\cdot)$, to transform a sample from a bounded uniform distribution to one from the required bounded Gaussian as follows:

$$u = \mathrm{U}(u \mid 0, 1) \tag{7.15}$$

$$\theta = \Phi^{-1}\Big(\Phi(b_{\min}) + u\big(\Phi(b_{\max}) - \Phi(b_{\min})\big)\Big) \tag{7.16}$$

An approach similar to this is used by Kotecha & Djurić [117]. The problem with this is that, for the Gaussian distribution, $\Phi(\cdot)$ and $\Phi^{-1}(\cdot)$ are not available analytically, so they must be approximated. There are well established algorithms for doing this, but to obtain a precise approximation requires a great deal of computation. Unfortunately, when the bounded region lies in the tail of a Gaussian distribution, $\Phi(b_{\max}) - \Phi(b_{\min})$ can become very small, so great precision is needed.

7.2.3.2 *Multivariate bounded Gaussians*

We have seen above that proposing from the unbounded distribution and simply rejecting samples outside the bounds is inefficient. It becomes even more so as the number of dimensions increases, as the bounded region becomes a progressively smaller proportion of the parameter space.

If equation (7.13) is used to transform the multivariate Gaussian into multiple independent ones, the cuboid forming the bounds must be transformed as well. Unless $\mathbf{C}$ is purely diagonal, this will result in a bounding region for ϕ with edges which are not parallel to the axes, so the bounds on each element of ϕ will be dependent on the values of the other elements, preventing independent sampling.
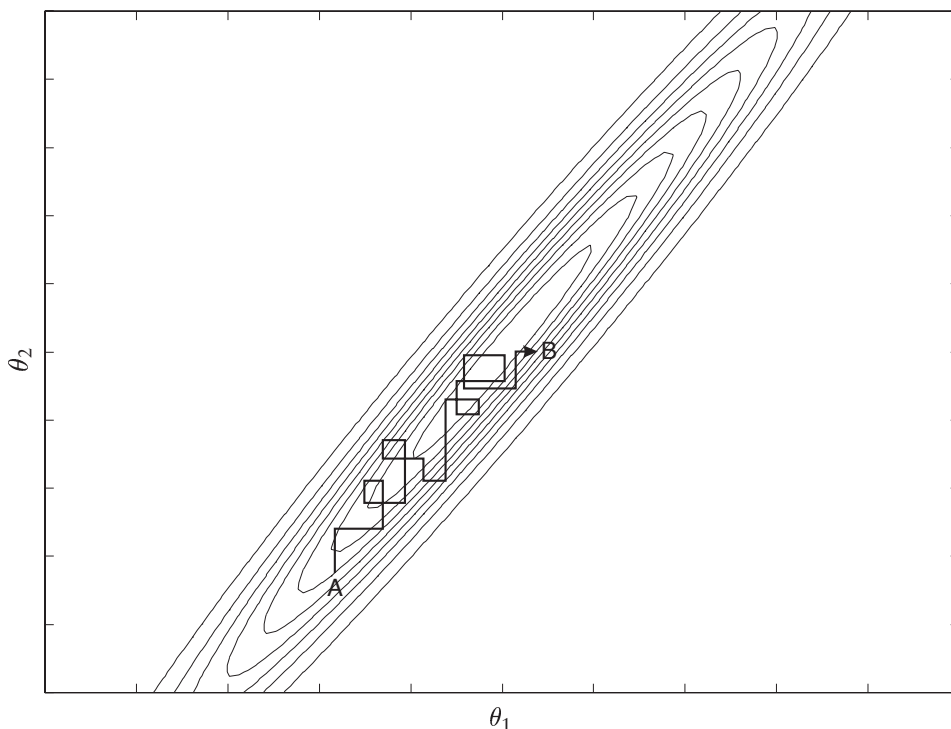
Figure 7.6. Contour map of the p.d.f. of a bivariate Gaussian distribution:
The heavy line shows a possible path for a Gibbs sampler to take between points A and B.

Gibbs sampling   One way to produce samples from $\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \mathbf{C})$ is to use a Gibbs sampler, in which each element is sampled from its full conditional distribution, in this case a truncated univariate Gaussian. This method is separately suggested by Geweke [70], Robert [162] and Kotecha & Djurić [117]. For this problem, it can be implemented straightforwardly by making the partition $(\cdot)_u$ in equation (7.6) contain a single element. Equation (7.11) then becomes the required full conditional distribution.

There is a potential drawback with this method: as discussed in §3.5.3.2, if, in an MCMC scheme, some variables are highly correlated, convergence will tend to be slow unless those variables are sampled jointly. Hence, if the off-diagonal elements of $\mathbf{C}$ are significant, convergence may be slow. Figure 7.6 illustrates the problem in two dimensions: the sampler can only move parallel to the axes, and will tend to stay in relatively high probability regions, so it cannot move between points A and B quickly. In higher dimensions, this lack of mobility becomes more serious.
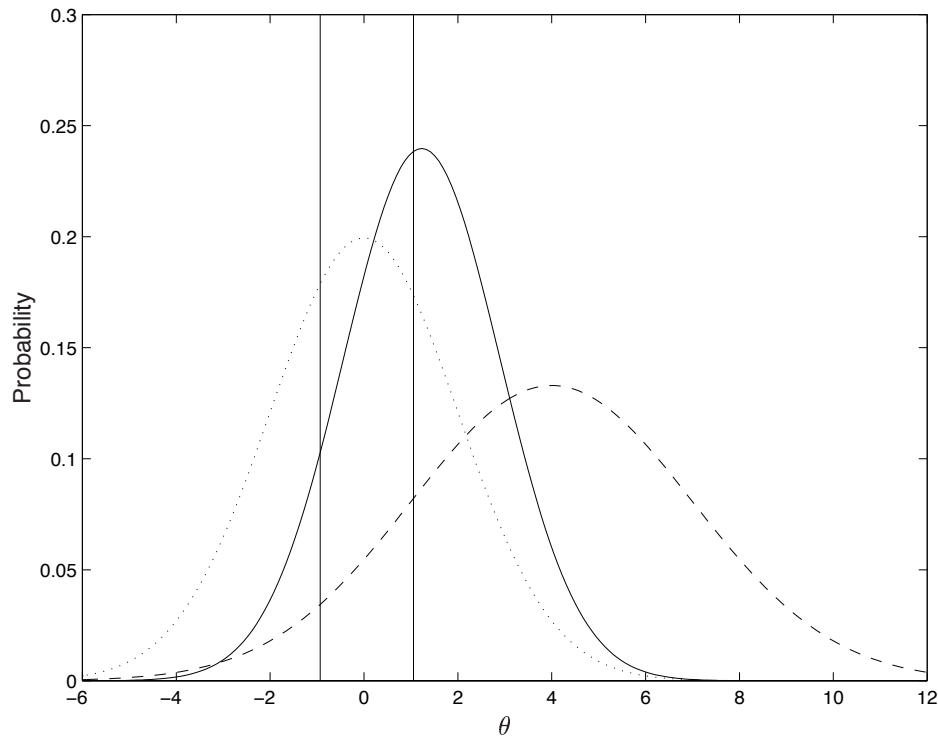
Figure 7.7. Gaussian windowing of a Gaussian distribution: (dashed) *the target Gaussian (without bounds),* (dotted) *the windowing Gaussian, and* (solid) *the resulting Gaussian from which samples will be drawn. The vertical lines show the bounds.*

Gaussian windowing   In order to be able to sample multiple components jointly, we now describe an alternative technique, which does not appear to have been suggested elsewhere.

As discussed earlier, direct rejection sampling using the unbounded distribution is very inefficient if most of the probability mass lies outside the bounds. Multiplying the (unbounded) target Gaussian distribution by another multivariate Gaussian, centred within the bounds, results in a distribution which is related to the target distribution but has a much greater probability mass within the bounds. This is illustrated, for the one-dimensional case, in Figure 7.7. Equation (A.1) shows that the combined distribution is itself a multivariate Gaussian:

$$\underbrace{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_c, \mathbf{C}_c)}_{\text{Combined}} \propto \underbrace{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \mathbf{C})}_{\substack{\text{Target} \\ \text{(unbounded)}}} \underbrace{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_w, \mathbf{C}_w)}_{\text{Window}} \qquad (7.17)$$

where $\boldsymbol{\mu}_w = \frac{\mathbf{b}_{max}+\mathbf{b}_{min}}{2}$, $\mathbf{C}_w = \kappa \operatorname{diag}(\mathbf{b}_{max} - \mathbf{b}_{min})$, $\mathbf{b}_{max}$ and $\mathbf{b}_{min}$ are vectors containing the upper and lower bounds on each element, κ is termed the *window factor*, and $\mathbf{C}_c$ and $\boldsymbol{\mu}_c$ can be determined as shown in §A.1.

Independent samples can be drawn from this combined distribution, and rejection sampling can be used to enforce the bounds. The bias introduced by the windowing function can then be removed by an independence Metropolis-Hastings sampler step (§3.3.1.2) with acceptance probability

$$\alpha(\boldsymbol{\theta} \to \boldsymbol{\theta}') = \min\left(1, \frac{\mathbf{N}(\boldsymbol{\theta}' \mid \boldsymbol{\mu}, \mathbf{C})}{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \mathbf{C})} \frac{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_c, \mathbf{C}_c)}{\mathbf{N}(\boldsymbol{\theta}' \mid \boldsymbol{\mu}_c, \mathbf{C}_c)}\right) \tag{7.18}$$

which simplifies through cancellation to

$$= \min\left(1, \frac{\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_w, \mathbf{C}_w)}{\mathbf{N}(\boldsymbol{\theta}' \mid \boldsymbol{\mu}_w, \mathbf{C}_w)}\right) \tag{7.19}$$

The tuning parameter, κ, varies the width of the windowing Gaussian and hence controls the trade-off between the acceptance rates in the rejection sampling step and in the independence sampler.

In general, the acceptance rate falls as the dimension of $\boldsymbol{\theta}$ increases, so for a given acceptance rate, the number of elements of $\mathbf{x}$ which can be sampled jointly will be limited. For the quantisation removal problem, experimentation has shown that joint sampling of five elements is acceptably efficient.

The structure of the covariance matrix (eq. 7.11) is such that nearer elements tend to have higher correlation, so sampling subblocks $\mathbf{x}_u$ of consecutive samples should give fastest convergence. If fixed subblocks were used, samples on the subblock boundaries would never be sampled jointly with those in the adjacent subblock. This problem can be avoided by applying a random offset to the subblocking on each iteration, as described in §7.2.6. Shephard & Pitt [172] use an alternative approach to random subblocking, which they call *stochastic knots*.

7.2.3.3   Comparison for synthetic AR data

The performance of the Gibbs sampling and Gaussian windowing methods was compared for a simple quantisation distortion problem. An AR(2) signal was synthesised, with poles at $0.99e^{j\pi\frac{\pm 15}{180}}$ and i.i.d. Gaussian excitation
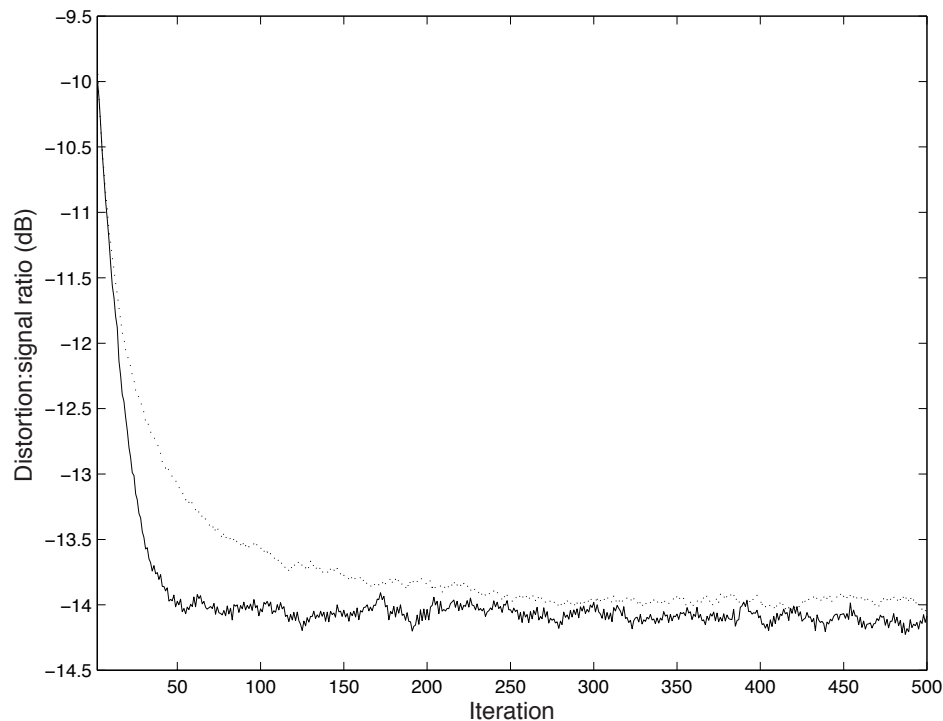
Figure 7.8. Comparison of bounded Gaussian sampling algorithms: (dotted) *univariate Gibbs sampler, and* (solid) *windowed Gaussian method, jointly sampling five consecutive components. The plot shows performance in reducing quantisation distortion in a synthetic AR(2) signal.*

of unit variance. The signal was quantised with a step height of 20, which introduced quantisation distortion 9.5 dB (r.m.s.) below the signal. Single blocks of 2048 samples were restored, with the AR model order fixed at 2. To avoid end effects without the need to change the conditioning from that described in §7.2.2 or to use exact likelihood methods, two samples before and after the block were made available from the unquantised signal. Fifty runs of 500 iterations were made with each of the algorithms, each run with a different block from the signal. In the windowing algorithm, κ was 10, and blocks of five consecutive samples were sampled jointly.

Figure 7.8 shows the mean distortion level across the 50 runs of each algorithm. It can be seen that the windowing algorithm exhibits much faster convergence. The total computation time was also significantly less than for the Gibbs sampling algorithm.

7.2.3.4   Discussion

On the basis of this test, the joint sampling algorithm will be used for the remainder of the experiments.

The efficiency might be improved further by changing the windowing function. A single Gaussian is not very close in shape to the required rectangular window. A closer fit could be achieved by using a mixture of Gaussians, for example equispaced Gaussians across the bounded region. A smaller value of κ could then be used, lowering the rejection rate. The drawback of this approach is the additional complexity involved in sampling from the mixture and evaluating its p.d.f. for use in the calculating the acceptance probability (see *e.g.* [30, 79]).

An alternative approach would be to correct from the windowed distribution to the desired distribution using rejection sampling. Rather than using a sufficiently large scaling factor for the envelope distribution to *dominate* the desired distribution (see §3.4.2), which could lead to very low acceptance rates, a lower value could be chosen and the discrepancy corrected through a Metropolis-Hastings step. This is called a *pseudo-dominating suggestion* [172], and forms a *rejection sampling chain* [182].

7.2.4   Blocking and overlap

As in §4.7 and Chapter 6, the signal model assumes stationarity, so time-varying audio signals must be processed in short blocks.

Since the AR model is being used to reconstruct the signal, the blocks must be contiguous and there must not be discontinuities at the boundaries. As in §4.7.5, this can be ensured by using the final k samples of the previous block as the initial conditions, $\mathbf{x}_0$, for the current block when sampling for k, $\mathbf{a}^{(k)}$ and σ_e^2. When sampling $\mathbf{x}$, k samples are used from either side of the interpolation point, regardless of block boundaries.[6]

---

[6]In the first and last blocks of the signal, these samples will not be available. This could be overcome by treating the end blocks as a special case for which the exact likelihood is used, together with one-sided conditioning in equation (7.11). The approach taken here is simply to use the values from $\mathbf{y}$, the unprocessed signal. The end-effects caused by this should be negligible when processing long audio signals.

7.2.5   Model mixing vs. Gibbs restoration

Estimating **x** from the sampler output uses all possible autoregressive models, weighted according to their posterior probability, to give a posterior mean estimate for **x**. Experience from audio interpolation applications, however, suggests that this may not be the best approach (see *e.g.* [146, 158]).

For an AR model with zero-mean excitation, the interpolant with maximum likelihood, which will also be the MAP estimate in this case, will be the one which minimises the excitation. This leads to the choice of interpolants which have excitation atypical of the surrounding signal—the excitation variance drops towards the centre of the interpolated region [87, Figure 5.8]. Using the value of the current interpolant at the final iteration will lead to a more typical signal with the excitation variance constant across the block. This "Gibbs restoration" [148] will be measurably worse in terms of mean squared error, but may sound better.

7.2.6   Algorithm

There are two parts to Algorithm 7.1: sampling **x** to reconstruct the signal and sampling the parameters of the AR model: k, $\mathbf{a}^{(k)}$, σ_a^2 and σ_e^2. The latter is done in the same manner as Algorithm 4.1. All variables and distributions within the b loop refer to the current block, which contains n_x samples. The value ν is a random offset applied to the subblocking (see p. 125); the **for** loop sets s to the starting point for each subblock.[7]

To allow direct comparison with the algorithm of §7.3.6, where the rest of the sampler is expensive to compute, **x** is sampled five times each iteration. Since they are relatively quick to compute, eight reversible-jump moves are proposed each iteration. For stand-alone use, better mixing would be produced if each step was performed fewer times each iteration.

For fast convergence, the initial values should be typical of the posterior distribution [73]. In these experiments, initial values of $\mathbf{a}^{(k)}$, σ_a^2 and σ_e^2 were generated using the quantised signal, **y**, by sampling from their full conditionals for a few iterations. The initial value of k was set arbitrarily; an atypical value was chosen in order to demonstrate the model selection capabilities of the method.

---

[7]To avoid clutter, Algorithm 7.1 is slightly simplified from that used in practise, which ensures that the first and last elements of **x** are sampled regardless of the value of ν.

Algorithm 7.1. Quantisation reduction using an AR model.

Choose initial values
for $i = \{1 \ldots \text{number of iterations}\}$
    for $b = \{1 \ldots \text{number of signal blocks}\}$
        — *(Sinusoidal moves (Algorithm 7.2) performed at this point)*
      for $g = \{1 \ldots \text{number of passes through } \mathbf{x} \text{ per iteration}\}$
        $v \sim \{0, 1, \ldots n_u\}$
        for $s = (v + 1)$ to n_x step n_u
            $u = \{s, \ldots s + n_u - 1\}$
            $\mathbf{x}_u \sim p(\mathbf{x}_u \mid \mathbf{x}_f, k, \mathbf{a}^{(k)}, \sigma_e^2, \mathbf{y})$
        end for
      end for
      for $r = \{1 \ldots \text{number of reversible-jump moves per iteration}\}$
        $k' \sim J(k' \mid k)$
        $z \sim U(0, 1)$
        if $z < \alpha(k \rightarrow k' \mid \sigma_a^2, \sigma_e^2, \mathbf{x})$
           $k = k'$
        end if
        $\mathbf{a}^{(k)} \sim p(\mathbf{a}^{(k)} \mid k, \sigma_a^2, \sigma_e^2, \mathbf{x})$
        $\sigma_a^2 \sim p(\sigma_a^2 \mid \mathbf{a}^{(k)})$
        $\sigma_e^2 \sim p(\sigma_e^2 \mid k, \mathbf{a}^{(k)}, \mathbf{x})$
      end for
    end for
end for

7.2.7   Results

7.2.7.1   *Audio signal*

The quantised version of the **piano** signal (Track 10, see §7.1.3) was split
into blocks of 1024 samples, and the sampler run for 200 iterations. Initial values were $\mathbf{x} = \mathbf{y}$ and $k = 6$; the remainder of the parameters were
drawn from their full conditionals. $k_{\max}$ was set to 50. A Monte Carlo estimate, $\hat{\mathbf{x}}$, of the signal was made using the final 100 iterations, in which the
distortion was reduced by an average of 9.9 dB (r.m.s.). As can be heard
on Track 12 on the accompanying CD (see Appendix C), this is a clearly
audible improvement.

    Figure 7.9 shows the signal together with, for each block, estimates of
$p(k \mid \mathbf{y})$ and $p(\sigma_e \mid \mathbf{y})$ and the distortion level before and after restoration.
Figure 7.10 shows part of the signal in which the quantisation distortion
was very noticeable. It can be seen that the error signal is much smaller
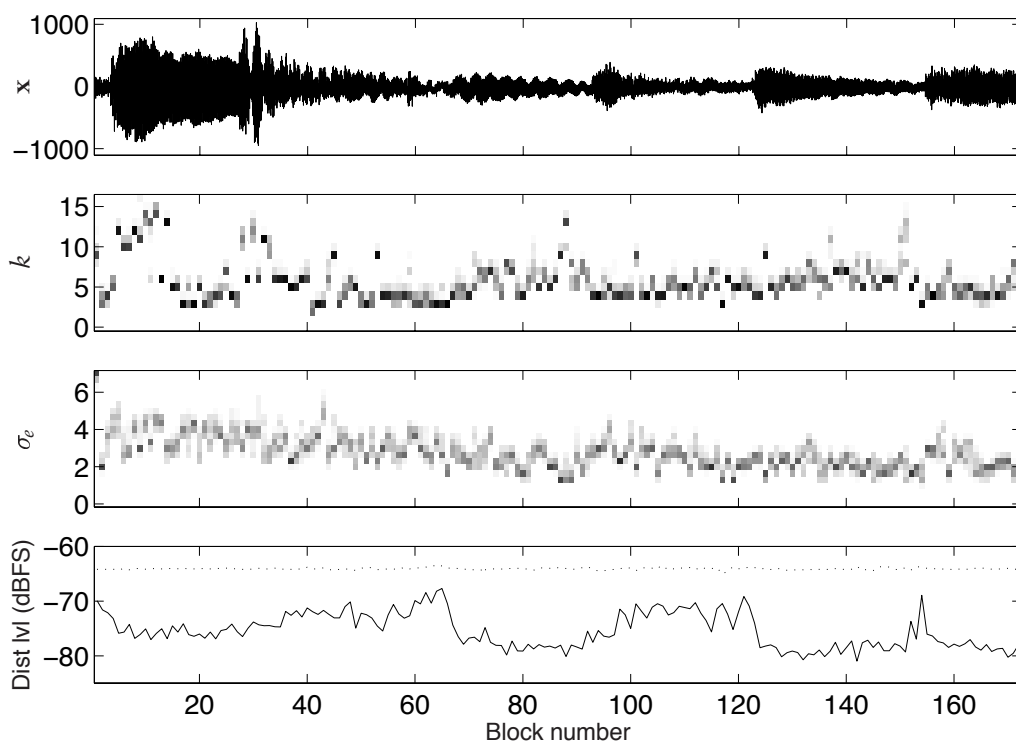
Figure 7.9. Restoration of quantised piano signal using plain AR model: *signal; histograms of estimated marginal posterior distributions (in which darkness represents probability); and distortion levels:* (dotted) **y**; (solid) **x̂**.
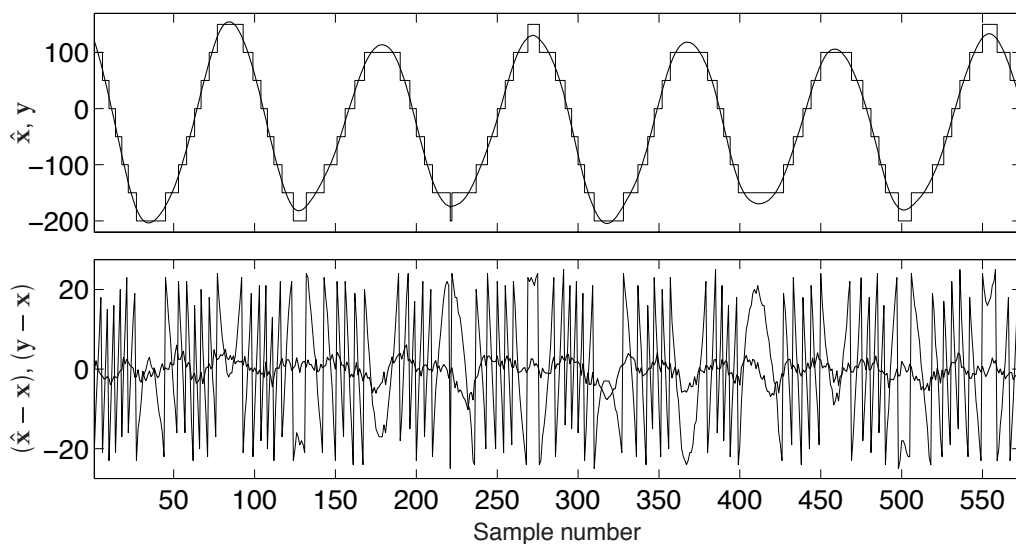
Figure 7.10. Part of block 80 from the run of Figure 7.9: (top) *signal and* (bottom) *error, in both of which the faint line corresponds to the quantised signal and the heavy line to the restored one.*
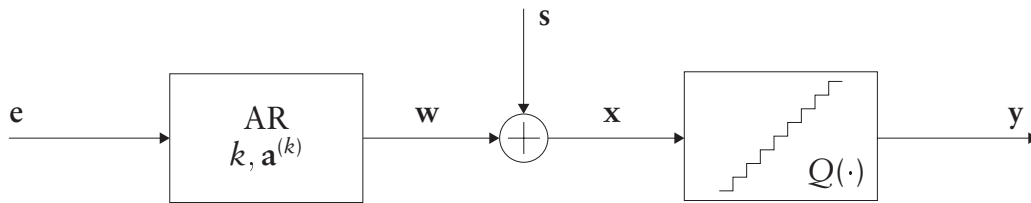
Figure 7.11. Modelling of quantised audio using sinusoidal + AR model: **y** *is the observed, quantised signal,* **x** *is the unquantised audio signal we aim to reconstruct,* **s** *is the sinusoidal component,* **w** *is the residual autoregressive component, and* **e** *is the excitation process.*

after restoration. For the part shown, which is typical of the quieter blocks, an improvement of 15 dB (r.m.s.) was achieved.

7.2.7.2 Gibbs restoration

As discussed in §7.2.5, in some audio applications, simply using the value of **x** from a single iteration has led to better sounding restorations. Track 13 is **x** from the final iteration of the run. It does not sound as good as Track 12, and it reduces distortion by only 7.6 dB (r.m.s.).

7.2.7.3 Effect of model order

Model orders $2 \leq k \leq 16$ were observed in the sampler output. Similar experiments, but using fixed model orders, were performed to show the importance of model selection.

If the signal was undermodelled by using a second order model throughout, the improvement was only 7.5 dB (r.m.s.) and noise can be heard in the restored signal (Track 15).

Overmodelling, by using a fixed order of 40 (Track 14), led to disturbing artefacts, similar to musical noise (§4.7.2), where some distortion elements were modelled as if part of the signal.

7.3   Sinusoids + AR model

As discussed in §7.1.3, the audio signals which exhibit the most noticeable quantisation distortion tend to be those which contain strong sinusoidal components. Hence, a possible improvement to the signal model is to model these deterministic components explicitly. Figure 7.11 shows a model which

does this. The residual, stochastic component, **w**, is modelled as autoregressive, as before:

$$\mathbf{x} = \mathbf{w} + \mathbf{s} \tag{7.20}$$

$$\mathbf{e} = \mathbf{A}\,\mathbf{w} = \mathbf{w}_1 - \mathbf{W}\,\mathbf{a} \tag{7.21}$$

A similar source model has previously been used for audio interpolation [87, §5.2.3] and in mixed-spectrum estimation (see *e.g.* [113]).

7.3.1 Sinusoidal model

7.3.1.1 *Basis function representation*

Within each block of the signal, the sinusoidal component, **s**, is modelled as a weighted sum of sinusoidal basis functions.[8] We use pairs of basis functions in quadrature to represent arbitrary phases:

$$g_t^{\|\langle i\rangle} = \sin\!\left(\omega^{\langle i\rangle}t\right) \qquad g_t^{\perp\langle i\rangle} = \cos\!\left(\omega^{\langle i\rangle}t\right) \tag{7.22}$$

If the weights, or *sinusoidal coefficients*, are contained in **c**, then this takes the form of the general linear model:

$$\mathbf{s} = \mathbf{G}\mathbf{c} \tag{7.23}$$

where **G** is formed from the basis functions:

$$\mathbf{G} = \begin{bmatrix} \uparrow & \uparrow & \cdots & \uparrow \\ \mathbf{g}^{\|\langle 1\rangle} & \mathbf{g}^{\perp\langle 1\rangle} & \cdots & \mathbf{g}^{\perp\langle\frac{n_c}{2}\rangle} \\ \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix} \tag{7.24}$$

7.3.1.2 *Choice of basis frequencies*

The choice of basis frequencies $\{\omega^{\langle i\rangle}\}$ and phases $\{\phi^{\langle i\rangle}\}$ can be made by a variety of means. We propose a method based on the subset selection techniques developed in Chapter 5:

- A list of candidate frequencies is drawn up. A quick way to do this is to perform a Hanning-windowed FFT on the quantised block and pick the $\frac{n_c}{2}$ frequencies with the highest energy.

---

[8]The sinusoidal components are estimated independently for each block. Hence all parameters mentioned here refer to only one block. For simplicity, this is not reflected in the notation.

- A binary indicator variable, γ_i, is associated with each pair of basis functions, such that when $\gamma_i = 0$, both basis functions with frequency $\omega^{\langle i \rangle}$ are removed from the model.

- The indicators, $\boldsymbol{\gamma}$, are treated as unknowns and incorporated into the posterior distribution.

This method constrains the sinusoids to match FFT bin frequencies in order to allow a fast subset selection algorithm to be used. Continuously variable frequencies could be allowed by including the frequencies in the model as unknown parameters. Andrieu & Doucet [7] use a mixed Metropolis-Hastings approach, consisting of independence sampler moves with proposals based on the FFT together with random-walk Metropolis moves to explore more locally. Djurić, Godsill, Fitzgerald & Rayner [51] present an alternative reversible-jump strategy, using predictive densities to allow the use of improper priors. In both cases, the goal is spectrum analysis, so accurate frequency estimation is important. This is obtained at the cost of much slower convergence, as the model is highly nonlinear in the frequency parameters. In the quantisation noise application, however, we are only interested in the reconstructed signal; any errors due to discrete frequencies will be incorporated into the AR model.

A similarly approximate approach has been used for audio interpolation [87, §5.2.3], but using a deterministic algorithm with a fixed number of sinusoids.

Picking frequencies from the FFT of the quantised signal seems a reasonable approach, as although the quantisation process adds many more sinusoidal components, it does not change the frequencies of those present in the original signal. It could, however, affect the amplitudes and phases, so these must be estimated as part of the restoration process.

7.3.2 Likelihood

The likelihood for the new signal model follows straightforwardly from equations (2.34), (7.20) & (7.23):

$$p\left(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}}, k, \mathbf{a}^{(k)}, \sigma_e^2\right) \approx p\left(\mathbf{x}_1 \mid \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}}, k, \mathbf{a}^{(k)}, \sigma_e^2, \mathbf{x}_0\right) \tag{7.25}$$

$$= p_{\mathbf{w}}\left(\mathbf{x} - \mathbf{G}_{\boldsymbol{\gamma}}\mathbf{c}_{\boldsymbol{\gamma}} \mid k, \mathbf{a}^{(k)}, \sigma_e^2\right) \tag{7.26}$$

$$= \mathbf{N}\left(\mathbf{A}^{(k)}\mathbf{x} - \mathbf{A}^{(k)}\mathbf{G}_{\boldsymbol{\gamma}}\mathbf{c}_{\boldsymbol{\gamma}} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\right) \tag{7.27}$$

where $\mathbf{c}_\gamma$ contains the parameters relating to basis functions currently included in the model.

7.3.3   Priors for new parameters

7.3.3.1   Prior for sinusoidal coefficients

In choosing a prior for the sinusoidal coefficients, $\mathbf{c}$, there are a number of aspects to consider:

- For subset selection to be possible, with an appropriate penalty for more complex models, a proper prior must be used (see §2.1.6.3).

- Since nothing is known about the phase of the signal, the priors should be circularly symmetric.

- A Gaussian prior, $p(c_i) = \mathrm{N}(c_i \mid 0, \sigma_c^2)$, has the advantage of being conjugate with the likelihood (eq. 7.27).

- Audio signals tend to exhibit a spectrum of shape $1/\omega$, *i.e.* equal energy per octave. Hence it might be reasonable for the variance of the priors on $\{c_i\}$ to be inversely proportional to $\{\omega^{\langle i \rangle}\}$.

- Since most instruments produce harmonic signals (see *e.g.* [201]), it is reasonable to expect the amplitudes of sinusoids at harmonically related frequencies to be related. Unfortunately this does not mean that the coefficients will be correlated, since the phases of the harmonics may differ.

For simplicity, we leave the $1/\omega$ prior for future research, and choose an i.i.d. Gaussian prior,

$$p(\mathbf{c}) = \mathrm{N}(\mathbf{c} \mid \mathbf{0}, \sigma_c^2 \mathbf{I}_{n_c}) \tag{7.28}$$

The prior variance could either be fixed or treated as a hyperparameter and estimated as part of the model. Experiments using an inverse Gamma approximation to the Jeffreys' prior (as used for σ_a^2) as a hyperprior were disappointing, with σ_c^2 tending to oscillate.

Therefore we choose to use a fixed value for σ_c^2 in each block. We have prior knowledge that the amplitude of any sinusoidal component is unlikely to be much larger than that of the signal. In the absence of any

other knowledge, the prior should have reasonably even support over this plausible range, and tail away outside it. We hence choose to make σ_c^2 proportional to the signal power in the block:

$$\sigma_c^2 = \frac{\zeta}{n_y} \mathbf{y}^T \mathbf{y} \qquad (7.29)$$

where ζ is a user-specified constant. A value $\zeta = 2$ has been found to be reasonable in experiments.

Although this is, strictly speaking, a data-dependent prior, it has the virtue of scale-independence—*i.e.* the prior will have the same effect on blocks containing the same signal at different levels. When modelling decaying notes, this seems a sensible requirement.

7.3.3.2   Prior for indicators

As in the previous subset selection problem (§5.4.4), independent Bernoulli priors are used for the indicators:

$$p(\gamma_i) = \upsilon \gamma_i + (1 - \upsilon)(1 - \gamma_i) \qquad (7.30)$$

such that $\upsilon = 1$ would force all sinusoids to be included and $\upsilon = 0$ would disable them. For current experiments, $\upsilon = 0.5$.

There is room for improvement here. The harmonic nature of the signal could be taken into account. For example, joint priors could be used for the indicators of harmonically related sinusoids, such that the presence of a sinusoid of frequency ω could increase the probability of inclusion of sinusoids at its harmonic frequencies $(2\omega, 3\omega, \dots)$ and at frequencies of which it is a harmonic $(\frac{1}{2}\omega, \frac{1}{3}\omega, \dots)$. Also, sinusoids are likely to persist over several blocks. This knowledge could be reflected by making the prior dependent on the indicator values in adjacent blocks.

7.3.4   Sampling the new parameters

The sinusoidal coefficients and associated indicators can be sampled in a very similar manner to the nonlinear model parameters and indicators in Chapter 5. Again, we expect the indicators and the associated parameters to be highly correlated, so they are sampled jointly. Since the basis functions are orthogonal, there is little advantage to sampling more than one indicator

at a time, so we partition such that γ_u is just one indicator, and $\mathbf{G}_u$ and $\mathbf{c}_u$ contain its two associated basis functions and parameters. If $\gamma_u = 1$ then $\mathbf{c}_{\gamma_u} = \mathbf{c}_u$ and $\mathbf{G}_{\gamma_u} = \mathbf{G}_u$, otherwise $\mathbf{c}_{\gamma_u}$ and $\mathbf{G}_{\gamma_u}$ are both empty.

The joint sampling is again performed by the composition of two steps:

$$\gamma_u \sim p(\gamma_u \mid \mathbf{x}, \boldsymbol{\gamma}_f, \mathbf{c}_{\boldsymbol{\gamma}_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2) \tag{7.31}$$

$$\mathbf{c}_{\gamma_u} \sim p(\mathbf{c}_{\gamma_u} \mid \mathbf{x}, \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2) \tag{7.32}$$

where the first step is not conditional on $\mathbf{c}_u$.

7.3.4.1 Sampling γ_u

The distribution required for step (7.31) can be obtained by marginalising $\mathbf{c}_{\gamma_u}$ from the full conditional for γ_u:

$$p(\gamma_u \mid \mathbf{x}, \boldsymbol{\gamma}_f, \mathbf{c}_{\boldsymbol{\gamma}_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$$
$$= \int p(\gamma_u \mid \mathbf{x}, \boldsymbol{\gamma}_f, \mathbf{c}_{\boldsymbol{\gamma}_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2) \, p(\mathbf{c}_{\gamma_u} \mid \sigma_c^2) \, d\mathbf{c}_{\gamma_u} \tag{7.33}$$

Using Bayes' theorem to express this in terms of the likelihood and priors, then performing the integral and neglecting terms which are independent of γ_u (see §B.3) gives

$$\propto p(\gamma_u) \frac{1}{\sigma_c^{n_{\mathbf{c}_{\gamma_u}}}} \sqrt{\left| \mathbf{C}_{cc_{\gamma_u}} \right|} \exp\left(\frac{1}{2\sigma_e^4} \mathbf{e}_f^T \mathbf{A}^{(k)} \mathbf{G}_{\gamma_u} \mathbf{C}_{cc_{\gamma_u}} \mathbf{G}_{\gamma_u}^T \mathbf{A}^{(k)T} \mathbf{e}_f \right) \tag{7.34}$$

where $n_{\mathbf{c}_{\gamma_u}}$ is the number of basis functions which are included, either zero or two, depending on the state of γ_u, and

$$\mathbf{e}_f = \mathbf{A}^{(k)}(\mathbf{x} - \mathbf{G}_{\boldsymbol{\gamma}_f} \mathbf{c}_{\boldsymbol{\gamma}_f}) \tag{7.35}$$

$$\mathbf{C}_{cc_{\gamma_u}} = \left(\frac{1}{\sigma_e^2} \mathbf{G}_{\gamma_u}^T \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \mathbf{G}_{\gamma_u} + \frac{1}{\sigma_c^2} \mathbf{I}_{n_{\mathbf{c}_{\gamma_u}}} \right)^{-1} \tag{7.36}$$

Hence step (7.31) can be accomplished by evaluating equation (7.34) (ρ_1) and the corresponding expression with $(\cdot)_u$ empty (ρ_0) and setting γ_u to one with probability $\frac{\rho_1}{\rho_0 + \rho_1}$, or to zero otherwise.

7.3.4.2 *Sampling* $\mathbf{c}_{\gamma_u}$

Sampling step (7.32) requires the full conditional distribution for $\mathbf{c}_{\gamma_u}$. This is derived in §B.3 from the likelihood and prior as

$$p\left(\mathbf{c}_{\gamma_u} \mid \mathbf{x}, \boldsymbol{\gamma}, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2\right) \propto \mathbf{N}\left(\mathbf{c}_{\gamma_u} \mid \boldsymbol{\mu}_{cc_{\gamma_u}}, \mathbf{C}_{cc_{\gamma_u}}\right) \tag{7.37}$$

where

$$\boldsymbol{\mu}_{cc_{\gamma_u}} = \tfrac{1}{\sigma_e^2} \mathbf{C}_{cc_{\gamma_u}} \mathbf{G}_{\gamma_u}^T \mathbf{A}^{(k)T} \mathbf{e}_f \tag{7.38}$$

Sampling from this is straightforward. It is also easy to sample jointly the whole of $\mathbf{c}_\gamma$, *i.e.* the coefficients of all the sinusoidal terms that are included in the model. To do this, the $(\cdot)_u$ partition is made to contain all n_γ active model terms, and $(\cdot)_k$ is left empty.

7.3.5 Blocking, overlap & conditioning

As before, time-varying audio signals are split into short blocks, within each of which they are treated as stationary. Since the FFT is performed on each block, computation is minimised by using block lengths which are a power of two. Blocks of length 1024 samples were used for the experiments described in §7.3.7.

It is necessary to enforce continuity across the block boundaries. For frequency domain signal processing methods, this can be achieved by using blocks with 50% overlap, and windowing the reconstructions with a window function which reduces to zero at the block boundaries. The output is then generated by summing the two windowed restorations of each sample and multiplying by a gain compensation function to ensure constant gain despite the windowing [87, §6.1]. This approach is also taken here, using Hanning windows.

As before, the signal is reconstructed by sampling $\mathbf{x}$ conditional on the model parameters, constrained to the region $\mathbf{Q}^{-1}(\mathbf{y})$. This is equivalent to sampling the AR component, $\mathbf{w}$, constrained to the region $\mathbf{Q}^{-1}(\mathbf{y}) - \mathbf{s}$, where $\mathbf{s}$ is the sinusoidal component.

As in §7.2.4, when sampling $\mathbf{w}$, k values from the previous block appear in the conditional likelihood as $\mathbf{w}_0$. Both $\mathbf{w}_0$ and the first k values of

Algorithm 7.2. Additional moves for AR + sinusoids model: *these moves are incorporated into Algorithm 7.1.*

> **for** $u = \{1 \dots \frac{n_c}{2}\}$
> $\quad \gamma_u \sim p(\gamma_u \mid \mathbf{x}, \boldsymbol{\gamma}_f, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$
> $\quad \mathbf{c}_{\gamma_u} \sim p(\mathbf{c}_{\gamma_u} \mid \mathbf{x}, \boldsymbol{\gamma}, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$
> **end for**
> $\mathbf{c}_{\gamma} \sim p(\mathbf{c}_{\gamma} \mid \mathbf{x}, \boldsymbol{\gamma}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$
> $\sigma_c^2 \sim p(\sigma_c^2 \mid \mathbf{c}_{\gamma})$

$\mathbf{w}$ after the end of the current block are used to provide boundary conditions when sampling $\mathbf{w}$, helping to ensure continuity of $\mathbf{x}$ across the block boundaries. Simply using the values of $\mathbf{w}$ from the adjacent blocks would result in discontinuities as (before the windowing procedure) $\mathbf{s}$ has sudden discontinuities at the block boundaries. Hence these values of $\mathbf{w}$ are provided by extending $\mathbf{s}$ from the current block to cover this extra range and subtracting it from $\mathbf{x}$ (which *is* continuous).

If the adjacent values of x_t are taken from the end of the last-but-one block and the beginning of the next-but-one block, *i.e.* from the blocks which abut the current one but do not overlap with it, then there is no interaction between the odd and even blocks, so they can be computed in parallel. This approach is similar to the *checker-board* updating scheme discussed by Roberts & Sahu [163]; their analysis suggests that it should not slow convergence.

7.3.6   Extended algorithm

Algorithm 7.2 shows the new sampling steps which are added to the algorithm of §7.2.6. The sampling steps for k, $\mathbf{a}^{(k)}$, σ_a^2 and σ_e^2 are unchanged from §7.2.1, except that now they refer to $\mathbf{w} = \mathbf{x} - \mathbf{s}$, rather than directly to $\mathbf{x}$. The steps of §7.2.2 require a simple modification for use in reconstructing $\mathbf{w}$: the bounds must now be offset by $-\mathbf{s}$, as each sample in the reconstructed signal is bounded to $(w_t + s_t) \in Q^{-1}(y_t)$.

Because sampling the indicators, $\boldsymbol{\gamma}$, each iteration is a relatively expensive process, several complete scans are made through $\mathbf{w}$ per iteration, so that it does not limit the convergence rate. In the experiments which follow, five scans are made per iteration.
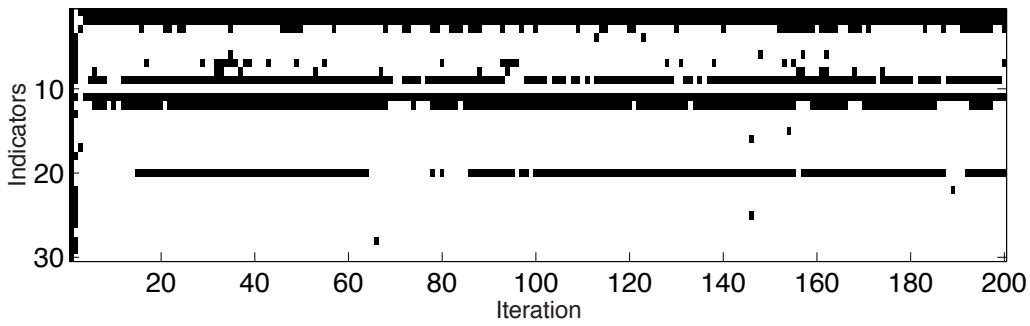
Figure 7.12. Evolution of γ in block 40: *black pixels represent $\gamma_i = 1$.*

7.3.7   Results

7.3.7.1   Audio signal

The new algorithm was applied to the same quantised signal as used in §7.2.7 (Track 10). It was again processed in blocks of length 1024 samples, but this time with 50% overlap. Thirty candidate frequencies were chosen in each block, so $n_c = 60$. Initially, all the indicators were set to one, so all 30 sinusoids were included. Other initial values were as before.

The sampler was again run for 200 iterations. Figure 7.12 shows the evolution of γ in a typical block over the course of the run. It can be seen that it converges quickly to a small number of sinusoids.

A Monte Carlo estimate, $\hat{x}$, of the signal was made using the final 100 iterations, in which the distortion was reduced by an average of 11 dB (r.m.s.), a slight improvement over the plain AR algorithm. It does not, however, sound noticeably different (Track 16).

Figure 7.13 shows the equivalent data to that in Figure 7.9, along with the estimated posterior distribution of the number of sinusoids. It can be seen that generally lower order AR models are used, with lower excitation variance, suggesting that the sinusoids have captured much of the structure of the signal.

Figure 7.14 shows the same part of the signal as Figure 7.10. Distortion is reduced by 17 dB (r.m.s.), 2 dB better than with the AR algorithm.

7.3.7.2   Model selection

To show that it is important not to use too many sinusoids, the sampler was rerun on the same signal with $v = 1$, so that all 30 candidate sinusoids
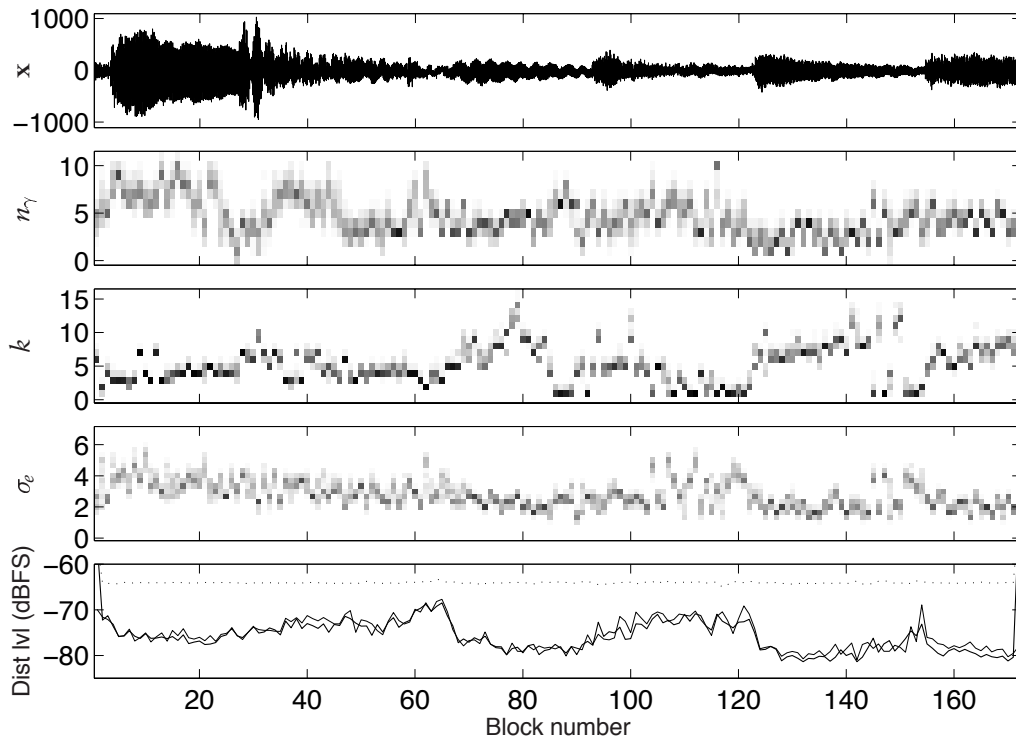
Figure 7.13. Restoration of quantised piano signal using sin+AR model: *signal; histograms of estimated marginal posterior distributions (in which darkness represents probability); and distortion levels:* (dotted) **y**; (heavy) **x̂**; (faint) *result from Figure 7.9 for comparison.*
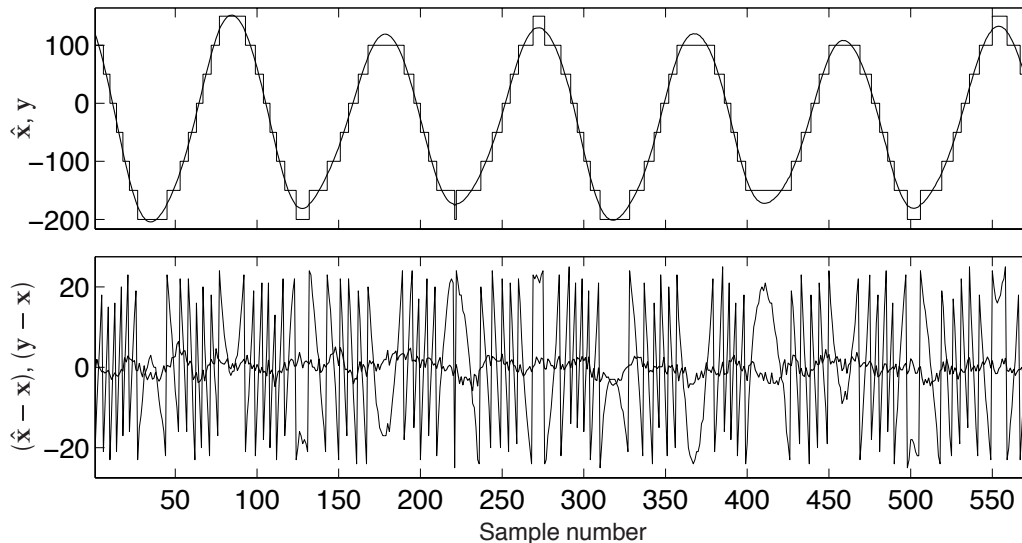
Figure 7.14. Part of block 80 from the run of Figure 7.13: (top) *signal and* (bottom) *error, in both of which the faint line corresponds to the quantised signal and the heavy line to the restored one.*

were included all the time. The performance seems very good: distortion was again reduced by 11 dB (r.m.s.). However, listening to the result (Track 17) reveals that there are now random, short-duration tones in the background, *i.e.* musical noise.

7.4 Discussion

This chapter has introduced quantisation distortion and the problems it can cause in audio recordings, then shown how they can easily be avoided by the use of dither and lamented the widespread ignorance of dither which has resulted in quantisation distortion marring many recordings.

It then considered the problem of restoring a coarsely quantised signal using an AR signal model. This required samples to be drawn from a truncated multivariate Gaussian distribution. Existing methods, based on univariate Gibbs sampler draws, were found to lead to slow convergence, so a new multivariate method was developed. Undermodelling and overmodelling were both found to significantly degrade performance.

Sinusoidal components were added to the model to attempt to improve performance, on the basis that it is signals with strong sinusoidal components which tend to produce noticeable quantisation distortion. Using too many sinusoids was found to produce musical noise; an approximate, but rapidly converging, subset selection method was developed to avoid this.

The sinusoid + AR model produces only slightly better restorations than the plain AR model, but has much potential for further improvement: block-to-block persistence can easily be incorporated into the prior structure, and harmonic relationships and $1/\omega$ spectral shapes are other possibilities.

Conclusions and further research 8

8.1 Conclusions

With the increase in available computing power it is becoming possible to address previously intractable audio restoration problems by using model-based statistical techniques.

This dissertation has explored the application of Bayesian techniques and Markov chain Monte Carlo methods to audio restoration problems, and made some contributions to MCMC model selection and sampling methods along the way.

In Chapters 4 & 5, it was found that exploiting the analytic properties of a model to design joint reversible-jump or Gibbs sampler moves can lead to much better mixing of the Markov chain, and hence faster convergence.

These new MCMC model order and subset uncertainty methods are straightforward to incorporate into MCMC frameworks for solving other problems—in §4.7, model order uncertainty was added to an existing noise reduction algorithm, and in Chapters 6 & 7 variable AR model orders were used in new algorithms.

In §4.7 and Chapter 7, it was shown to be important to the avoidance of audible artefacts to allow model orders to vary over the duration of an audio signal.

In Chapter 6, a previously-proposed cascade AR-NAR model for non-linearly distorted audio was reimplemented in a fully Bayesian manner. The MCMC methods of Chapters 4 & 5 were used to allow for model order uncertainty in the linear model and to perform subset selection in the non-linear model,

The approach was then extended such that long, time-varying audio signals could be processed, with nonlinear model estimation being performed *jointly* across all the blocks.

In Chapter 7, the problem of restoring quantised audio was addressed for the first time, with promising results. This was greatly accelerated by the development of a new method for drawing samples from a truncated multivariate Gaussian distribution. The new method results in significantly faster convergence than Gibbs sampler-based methods when there is strong correlation between the different components.

8.2   Suggestions for further research

8.2.1   Model order uncertainty

The reversible-jump model order sampling approach of Chapter 4 does not currently enforce the stability of the AR model. This has not proved to be a problem when the algorithm was used in §4.7 or Chapters 6 & 7, but could be in other applications.

Whilst it is simple to use rejection sampling to avoid unstable parameter values, there is a problem with the marginalisation of $\mathbf{a}^{(k)}$ when calculating the acceptance probability—it is now necessary to integrate only over the stable region, which is difficult. However, the only effect on the acceptance probability is to scale the numerator by $\omega(k', \sigma_a^2)$ and the denominator by $\omega(k, \sigma_a^2)$. Although ω is difficult to calculate, it is simply a scalar and a function only of the model order, which is discrete and bounded, and one other variable. Hence it might be practicable to precalculate it for each value of k and a range of values of σ_a^2 (in which it is probably smooth), and use these to produce interpolated values for use in the simulation.

With stability enforced, it would then be straightforward to incorporate the $p(\mathbf{x}_0 \mid k, \mathbf{a}^{(k)}, \sigma_e^2)$ terms necessary to use the exact likelihood.

Another possible enhancement would be to relax the i.i.d. Gaussian assumption on the excitation. Sinusoidal or pulse-train excitation could be incorporated using the techniques of §7.3, or impulses could be allowed using indicators in a sampling scheme similar to that of Chapter 5 [88].

8.2.2   Nonlinearly distorted audio signals

As discussed in Chapter 6, although the AR-NAR model selection scheme and restoration algorithm works well on audio data which has been distorted by an NAR process, it does not significantly improve any of the

real-world distorted audio signals on which it has been tried. This may be because much larger NAR models are needed than are practicable with current computing hardware, or because an NAR process is not a good model of real distortion-causing systems.

Before writing off the NAR model all together, it might be worthwhile to perform input/output NAR modelling of typical audio equipment, such as tape machines and disc cutting systems. Time-alignment could cause difficulty here, as record and replay speeds are never completely steady.

The model could be improved by incorporating specialised nonlinear terms based on physical models of the distortion-causing process. For example, the geometry which causes *tracing distortion* in records is well understood [42, 43, 123, 159, 185] and can be readily simulated; parameter estimation (linear velocity and cutting and playback stylus radii) is, however, difficult, as the model is no longer linear-in-the-parameters.

An extra, linear stage—either AR or MA—could be added, after the nonlinear stage, to model the linear response of the playback mechanism. Both tape and record players include equalisation in the playback circuitry. At present, this must be modelled as part of the NAR model; separating it out could allow a smaller, simpler NAR stage. With this three stage model (first suggested by Mercer [141]), it might be possible to replace the NAR stage by a memoryless nonlinearity for simple problems such as optical soundtracks.

Noise added to the observed signal could prevent the AR-NAR algorithm from finding the right NAR model. Incorporating white Gaussian observation noise explicitly might help. Impulsive noise could also be allowed, as discussed in §8.2.1.

The technique developed for joint estimation of a channel model over many blocks of time-varying audio data (§6.5.1) could be useful even with purely linear channel models, for applications such as reverberation modelling and removal.

8.2.3   Noise reduction

The artefacts of the model-based noise reduction process could be further reduced, as suggested in §4.7.7, by introducing inter-block dependence into the priors on the model order and possibly the parameter values and excitation variance.

A more elaborate approach would be to use reversible-jump change-

point detection methods (see *e.g.* [177]) to adapt the block lengths and placements so that abrupt changes in the signal always occurred on block boundaries. Much greater inter-block smoothness could then be enforced where change-points were not found.

Both in noise reduction and quantisation removal applications, when the model order is too high, the AR model tends to position poles such that they model high frequency noise or distortion components. Audio signals tend to exhibit a $1/\omega$ spectral shape; incorporating this knowledge into the prior on $\mathbf{a}^{(k)}$ might be beneficial. This may be possible, while retaining the multivariate Gaussian prior structure, through an adaptation of the smoothness priors of Kitagawa & Gersch [114].

It would be simple to incorporate the sinusoidal signal components of §7.3 into the noise reduction framework, which might lead to improved performance on signals with strong sinusoidal components.

8.2.4   Quantisation distortion

Although approximate, the sinusoidal model of §7.3 has the advantage that it would be relatively easy to incorporate inter-block dependence into the sinusoidal indicators, γ, such that persistent sinusoids are encouraged but those which appear in only a single block, which are more likely to be musical noise, are discouraged.

It would be interesting to see whether full reversible-jump sinusoidal estimation schemes (such as [7] or even [201]) improve the performance of the quantisation removal algorithm (at the expense of speed).

The windowing method for sampling truncated Gaussians could be modified, as suggested in §7.2.3.4, to use rejection sampling chains [182]. This may allow more components to be sampled jointly, and hence faster convergence. Another approach which should be tried is the *Geweke-Hajivassiliou-Keane simulator* [96], which again uses independence sampling, but with a different proposal distribution.

Quantisation distortion is generally most noticeable in the quiet parts of a signal, when the signal is quite near the noise floor. Performing noise reduction jointly with quantisation removal could therefore result in better restorations.

As discussed in §7.1.4.1, dither is essentially added noise. It would be a very challenging task to try to improve signals which have been correctly

dithered to a low resolution—although the statistics of the dither component may be known precisely, the dither is usually non-Gaussian, and so would be difficult to integrate out.

8.2.5 Clipped signals

To restore a clipped signal, those parts of the signal which exceed the clipping threshold must be interpolated. This is a very similar problem to that of Chapter 7: the interpolant must be constrained to be above the clipping threshold, so draws must be made from a multivariate Gaussian distribution which is truncated on one side.

The problem differs from quantisation removal in that quite long sections may need to be interpolated—possibly many tens of samples. Hence the deterministic component provided by sinusoidal modelling is likely to prove quite important.

Signals which have been clipped in the analogue domain may exhibit gentle saturation before they reach the clipping level. It might therefore be sensible to incorporate a memoryless nonlinearity into the model before the clipping stage.

8.2.6 Other suggestions

It has become very clear in this research that r.m.s. distortion measurements do not correlate at all well with human perception. Although there has been much research effort put into psychoacoustic modelling for perceptual coding schemes (see *e.g.* [41]), no simple measure is widely used to objectively evaluate distortion.

Bayes' theorem (eq. 2.3) is a model of a learning process, and hence Bayesian methods are highly suited to sequential learning, in which Bayes' theorem can be applied repeatedly to update parameter estimates on the basis of each new sample. Sequential Bayesian estimation using MCMC (see *e.g.* [129]) is likely to be a growth area. The noise reduction, quantisation removal and declipping algorithms could all be reimplemented in such a manner to provide useful real-time tools once sufficient computing power becomes available.

Manipulation of Gaussians

<div style="text-align: right; font-size: 2em;">A</div>

This appendix presents some useful results for the manipulation of multivariate Gaussian distributions.

A.1 Product of Gaussians

If a conjugate Gaussian prior is used in a model with Gaussian likelihood, the expression for posterior will include a product of these two Gaussians. For the univariate case, Box & Tiao [24, §A1.1] show that this product is proportional to a single Gaussian; we state here the general, multivariate result, without neglecting constants of proportionality, as these prove important in model order selection (Chapter 4) and subset selection (Chapter 5). It can be derived straightforwardly by multiplying out and completing the square. The parameter vector, $\boldsymbol{\theta}$, is of dimension n.

$$\mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_1, \mathbf{C}_1) \times \mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_2, \mathbf{C}_2) = \Upsilon_c \times \mathbf{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}_c, \mathbf{C}_c) \tag{A.1}$$

where

$$\mathbf{C}_c = \left(\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1}\right)^{-1} \tag{A.2}$$

$$\boldsymbol{\mu}_c = \mathbf{C}_c\left(\mathbf{C}_1^{-1}\boldsymbol{\mu}_1 + \mathbf{C}_2^{-1}\boldsymbol{\mu}_2\right) \tag{A.3}$$

$$\Upsilon_c = (2\pi)^{-\frac{n}{2}} \frac{|\mathbf{C}_c|^{\frac{1}{2}}}{|\mathbf{C}_1|^{\frac{1}{2}} |\mathbf{C}_2|^{\frac{1}{2}}} \exp\left(-\tfrac{1}{2}\left[\boldsymbol{\mu}_1^T\mathbf{C}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^T\mathbf{C}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_c^T\mathbf{C}_c^{-1}\boldsymbol{\mu}_c\right]\right)$$
$$\tag{A.4}$$

This identity extends in an obvious manner to products of larger numbers of Gaussians.

A.2 *Linear transformation of a Gaussian*

If a linear transformation of the random vector $\boldsymbol{\theta}$ obeys an i.i.d. Gaussian distribution, then $p(\boldsymbol{\theta})$ can be expressed as a multivariate Gaussian distribution:

$$\mathbf{N}\big(\mathbf{B}\boldsymbol{\theta} \mid \boldsymbol{\mu}, \sigma^2 \mathbf{I}_m\big) = \Upsilon_l \times \mathbf{N}\big(\boldsymbol{\theta} \mid (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\boldsymbol{\mu}, \sigma^2(\mathbf{B}^T\mathbf{B})^{-1}\big) \qquad (\text{A.5})$$

where

$$\Upsilon_l = (2\pi\sigma^2)^{\frac{n-m}{2}} \left|\mathbf{B}^T\mathbf{B}\right|^{-\frac{1}{2}} \exp\big(-\tfrac{1}{2\sigma^2}\big[\boldsymbol{\mu}^T\boldsymbol{\mu} - \boldsymbol{\mu}^T\mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\boldsymbol{\mu}\big]\big) \qquad (\text{A.6})$$

and n and m are the dimensions of $\boldsymbol{\theta}$ and $\mathbf{B}\boldsymbol{\theta}$, respectively. This can be verified by multiplying out and matching terms.

Derivation of posterior distributions

B

This appendix outlines some of the derivations required in the text. In each of the three sections, the same basic manipulations are performed, but we go into some detail in order to avoid ambiguity and to aid reimplementation.

B.1 Marginal posterior for reversible-jump moves

The expression for the acceptance probability in §4.4.2 requires knowledge of the following distribution, up to a constant of proportionality:[1]

$$p(k' \mid \mathbf{x}, \mathbf{a}_f, \sigma_a^2, \sigma_e^2) \propto p(k') \int \underbrace{p(\mathbf{x} \mid k', \mathbf{a}, \sigma_e^2)}_{\text{Likelihood}} p(\mathbf{a}_u \mid \sigma_a^2) \, d\mathbf{a}_u \qquad \text{(B.1)}$$

The approximate likelihood (eq. 4.13) can be rewritten as

$$p\left(\mathbf{x} \mid k', \mathbf{a}, \sigma_e^2\right) \approx \mathbf{N}\left(\mathbf{x}_1 - \mathbf{X}_f \mathbf{a}_f - \mathbf{X}_u \mathbf{a}_u \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\right) \qquad \text{(B.2)}$$

$$= \mathbf{N}\left(\mathbf{X}_u \mathbf{a}_u \mid \underbrace{\mathbf{x}_1 - \mathbf{X}_f \mathbf{a}_f}_{\triangleq \, \mathbf{e}_f}, \sigma_e^2 \mathbf{I}_{n_e}\right) \qquad \text{(B.3)}$$

such that $\mathbf{e}_f$ is the excitation signal which would need to be applied to an AR model containing only those terms whose parameters are included in $\mathbf{a}_f$. Using the identity of §A.2, this Gaussian can be rearranged as

$$p\left(\mathbf{x} \mid k', \mathbf{a}, \sigma_e^2\right) = \Upsilon_{l\mathbf{a}_u} \mathbf{N}\left(\mathbf{a}_u \mid \boldsymbol{\mu}_{l\mathbf{a}_u}, \mathbf{C}_{l\mathbf{a}_u}\right) \qquad \text{(B.4)}$$

---

[1]For clarity, the $(\cdot)^{(k')}$ notation is neglected in this section, as all instances of $\mathbf{a}$ and $\mathbf{X}$ are associated with the model of order k'.

where

$$\mathbf{C}_{la_u} = \sigma_e^2 \left(\mathbf{X}_u^T \mathbf{X}_u \right)^{-1} \tag{B.5}$$

$$\boldsymbol{\mu}_{la_u} = \left(\mathbf{X}_u^T \mathbf{X}_u \right)^{-1} \mathbf{X}_u^T \mathbf{e}_f \tag{B.6}$$

$$\Upsilon_{la_u} = \frac{\exp\left(-\frac{1}{2\sigma_e^2} \left[\mathbf{e}_f^T \mathbf{e}_f - \mathbf{e}_f^T \mathbf{X}_u (\mathbf{X}_u^T \mathbf{X}_u)^{-1} \mathbf{X}_u^T \mathbf{e}_f \right] \right)}{(\sqrt{2\pi}\sigma_e)^{n_e - n_u} \sqrt{|\mathbf{X}_u^T \mathbf{X}_u|}} \tag{B.7}$$

and n_u is the number of terms in $\mathbf{a}_u$.

Now that the likelihood is in the convenient form of a Gaussian in $\mathbf{a}_u$, it can be multiplied by the prior on $\mathbf{a}_u$ (eq. 4.15) to form the integrand of equation (B.1):

$$p(\mathbf{x}, \mathbf{a}_u \mid k', \mathbf{a}_f, \sigma_a^2, \sigma_e^2) = p(\mathbf{x} \mid k', \mathbf{a}, \sigma_e^2)\, p(\mathbf{a}_u \mid \sigma_a^2) \tag{B.8}$$

$$= \Upsilon_{la_u} \mathbf{N}\left(\mathbf{a}_u \mid \boldsymbol{\mu}_{la_u}, \mathbf{C}_{la_u} \right) \mathbf{N}\left(\mathbf{a}_u \mid 0, \mathbf{C}_{pa_u} \right) \tag{B.9}$$

Since the prior is conjugate, this product can be rearranged in the form of a single Gaussian:

$$= \Upsilon_{ca_u} \mathbf{N}\left(\mathbf{a}_u \mid \boldsymbol{\mu}_{ca_u}, \mathbf{C}_{ca_u} \right) \tag{B.10}$$

where

$$\mathbf{C}_{ca_u} = \left(\mathbf{C}_{la_u}^{-1} + \mathbf{C}_{pa_u}^{-1} \right)^{-1} \tag{B.11}$$

$$\boldsymbol{\mu}_{ca_u} = \mathbf{C}_{ca_u} \mathbf{C}_{la_u}^{-1} \boldsymbol{\mu}_{la_u} \tag{B.12}$$

$$\Upsilon_{ca_u} = \frac{\Upsilon_{la_u}}{\sqrt{2\pi}^{n_u}} \sqrt{\frac{|\mathbf{C}_{ca_u}|}{|\mathbf{C}_{la_u}||\mathbf{C}_{pa_u}|}}$$
$$\times \exp\left(-\frac{1}{2} \left[\boldsymbol{\mu}_{la_u}^T \mathbf{C}_{la_u}^{-1} \boldsymbol{\mu}_{la_u} - \boldsymbol{\mu}_{ca_u}^T \mathbf{C}_{ca_u}^{-1} \boldsymbol{\mu}_{ca_u} \right] \right) \tag{B.13}$$

which simplify, after much cancellation, to

$$\mathbf{C}_{ca_u} = \left(\frac{1}{\sigma_e^2} \mathbf{X}_u^T \mathbf{X}_u + \mathbf{C}_{pa_u}^{-1} \right)^{-1} \tag{B.14}$$

$$\boldsymbol{\mu}_{ca_u} = \frac{1}{\sigma_e^2} \mathbf{C}_{ca_u} \mathbf{X}_u^T \mathbf{e}_f \tag{B.15}$$

$$\Upsilon_{ca_u} = \frac{\sqrt{|\mathbf{C}_{ca_u}|}}{(\sqrt{2\pi}\sigma_e)^{n_e} \sqrt{|\mathbf{C}_{pa_u}|}} \exp\left(-\frac{1}{2\sigma_e^2} \left[\mathbf{e}_f^T \mathbf{e}_f - \frac{1}{\sigma_e^2} \mathbf{e}_f^T \mathbf{X}_u \mathbf{C}_{ca_u} \mathbf{X}_u^T \mathbf{e}_f \right] \right) \tag{B.16}$$

Now equation (B.1) can be written in the form

$$p(k' \mid \mathbf{x}, \mathbf{a}_f, \sigma_a^2, \sigma_e^2) \propto p(k') \Upsilon_{c\mathbf{a}_u} \int \mathbf{N}\big(\mathbf{a}_u \mid \boldsymbol{\mu}_{c\mathbf{a}_u}, \mathbf{C}_{c\mathbf{a}_u}\big) \, d\mathbf{a}_u \qquad \text{(B.17)}$$

but the unbounded integral is equal to unity, so

$$\propto p(k') \Upsilon_{c\mathbf{a}_u} \qquad \text{(B.18)}$$

B.2 Derivation of marginal posterior for β_u

In §5.5.2.1, it is necessary to sample from the distribution

$$p(\beta_u \mid \mathbf{y}, \beta_f, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2) \propto p(\beta_u) \int \underbrace{p(\mathbf{y} \mid \beta, \mathbf{b}_\beta, \sigma_e^2)}_{\text{Likelihood}} p(\mathbf{b}_{\beta_u} \mid \sigma_b^2) \, d\mathbf{b}_{\beta_u} \quad \text{(B.19)}$$

The steps in this derivation are very similar to those required in §B.1 to find the acceptance probability for reversible-jump moves when the parameters are proposed from their full conditional distributions.

The approximate likelihood (eq. 5.17) can be rewritten as

$$p\big(\mathbf{y} \mid \beta, \mathbf{b}_\beta, \sigma_e^2\big) \approx \mathbf{N}\big(\mathbf{y}_1 - \mathbf{Y}_{\beta_f}\mathbf{b}_{\beta_f} - \mathbf{Y}_{\beta_u}\mathbf{b}_{\beta_u} \mid \mathbf{0}, \sigma_e^2 \mathbf{I}_{n_e}\big) \qquad \text{(B.20)}$$

$$= \mathbf{N}\big(\mathbf{Y}_{\beta_u}\mathbf{b}_{\beta_u} \mid \underbrace{\mathbf{y}_1 - \mathbf{Y}_{\beta_f}\mathbf{b}_{\beta_f}}_{\triangleq \, \mathbf{e}_f}, \sigma_e^2 \mathbf{I}_{n_e}\big) \qquad \text{(B.21)}$$

such that $\mathbf{e}_f$ is the excitation signal which would need to be applied to the AR model if $\mathbf{x}$ was reconstructed from $\mathbf{y}$ using only the NAR model terms in the partition $(\cdot)_{\beta_f}$.

After some manipulation, this Gaussian can be rearranged as

$$p\big(\mathbf{y} \mid \beta, \mathbf{b}_\beta, \sigma_e^2\big) = \Upsilon_{l\mathbf{b}_{\beta_u}} \mathbf{N}\big(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{l\mathbf{b}_{\beta_u}}, \mathbf{C}_{l\mathbf{b}_{\beta_u}}\big) \qquad \text{(B.22)}$$

where

$$\mathbf{C}_{lb_{\beta_u}} = \sigma_e^2 \left(\mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u}\right)^{-1} \tag{B.23}$$

$$\boldsymbol{\mu}_{lb_{\beta_u}} = \left(\mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u}\right)^{-1} \mathbf{Y}_{\beta_u}^T \mathbf{e}_f \tag{B.24}$$

$$\Upsilon_{lb_{\beta_u}} = \frac{\exp\left(-\frac{1}{2\sigma_e^2}\left[\mathbf{e}_f^T \mathbf{e}_f - \mathbf{e}_f^T \mathbf{Y}_{\beta_u}\left(\mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u}\right)^{-1}\mathbf{Y}_{\beta_u}^T \mathbf{e}_f\right]\right)}{\left(\sqrt{2\pi}\sigma_e\right)^{n_e - n_{\beta_u}} \sqrt{|\mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u}|}} \tag{B.25}$$

and n_{β_u} is the number of ones in $\boldsymbol{\beta}_u$.

Now that the likelihood is in the convenient form of a Gaussian in $\mathbf{b}_{\beta_u}$, it can be multiplied by the prior on $\mathbf{b}_{\beta_u}$ (eq. 5.19) to form the integrand of equation (B.19):

$$p(\mathbf{y}, \mathbf{b}_{\beta_u} \mid \boldsymbol{\beta}, \mathbf{b}_{\beta_f}, \sigma_b^2, \sigma_e^2)$$

$$= p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{b}_\beta, \sigma_e^2)\, p(\mathbf{b}_{\beta_u} \mid \sigma_b^2) \tag{B.26}$$

$$= \Upsilon_{lb_{\beta_u}} \mathbf{N}\left(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{lb_{\beta_u}}, \mathbf{C}_{lb_{\beta_u}}\right) \mathbf{N}\left(\mathbf{b}_{\beta_u} \mid 0, \mathbf{C}_{pb_{\beta_u}}\right) \tag{B.27}$$

Since the prior is conjugate, this product can be rearranged in the form of a single Gaussian:

$$= \Upsilon_{cb_{\beta_u}} \mathbf{N}\left(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{cb_{\beta_u}}, \mathbf{C}_{cb_{\beta_u}}\right) \tag{B.28}$$

where

$$\mathbf{C}_{cb_{\beta_u}} = \left(\mathbf{C}_{lb_{\beta_u}}^{-1} + \mathbf{C}_{pb_{\beta_u}}^{-1}\right)^{-1} \tag{B.29}$$

$$\boldsymbol{\mu}_{cb_{\beta_u}} = \mathbf{C}_{cb_{\beta_u}} \mathbf{C}_{lb_{\beta_u}}^{-1} \boldsymbol{\mu}_{lb_{\beta_u}} \tag{B.30}$$

$$\Upsilon_{cb_{\beta_u}} = \frac{\Upsilon_{lb_{\beta_u}}}{\sqrt{2\pi}^{n_{\beta_u}}} \sqrt{\frac{|\mathbf{C}_{cb_{\beta_u}}|}{|\mathbf{C}_{lb_{\beta_u}}||\mathbf{C}_{pb_{\beta_u}}|}}$$

$$\times \exp\left(-\tfrac{1}{2}\left[\boldsymbol{\mu}_{lb_{\beta_u}}^T \mathbf{C}_{lb_{\beta_u}}^{-1} \boldsymbol{\mu}_{lb_{\beta_u}} - \boldsymbol{\mu}_{cb_{\beta_u}}^T \mathbf{C}_{cb_{\beta_u}}^{-1} \boldsymbol{\mu}_{cb_{\beta_u}}\right]\right) \tag{B.31}$$

which simplify, after much cancellation, to

$$\mathbf{C}_{cb_{\beta_u}} = \left(\tfrac{1}{\sigma_e^2} \mathbf{Y}_{\beta_u}^T \mathbf{Y}_{\beta_u} + \mathbf{C}_{pb_{\beta_u}}^{-1} \right)^{-1} \tag{B.32}$$

$$\boldsymbol{\mu}_{cb_{\beta_u}} = \tfrac{1}{\sigma_e^2} \mathbf{C}_{cb_{\beta_u}} \mathbf{Y}_{\beta_u}^T \mathbf{e}_f \tag{B.33}$$

$$\Upsilon_{cb_{\beta_u}} = \frac{\sqrt{|\mathbf{C}_{cb_{\beta_u}}|}}{\left(\sqrt{2\pi}\sigma_e\right)^{n_e} \sqrt{|\mathbf{C}_{pb_{\beta_u}}|}} \exp\left(-\tfrac{1}{2\sigma_e^2} \left[\mathbf{e}_f^T \mathbf{e}_f - \tfrac{1}{\sigma_e^2} \mathbf{e}_f^T \mathbf{Y}_{\beta_u} \mathbf{C}_{cb_{\beta_u}} \mathbf{Y}_{\beta_u}^T \mathbf{e}_f \right] \right) \tag{B.34}$$

Now equation (B.19) can be written in the form

$$p(\boldsymbol{\beta}_u \mid \mathbf{y}, \boldsymbol{\beta}_f, \mathbf{b}_{\beta_f}, k, \mathbf{a}^{(k)}, \sigma_b^2, \sigma_e^2)$$

$$\propto p(\boldsymbol{\beta}_u) \, \Upsilon_{cb_{\beta_u}} \int \mathbf{N}\!\left(\mathbf{b}_{\beta_u} \mid \boldsymbol{\mu}_{cb_{\beta_u}}, \mathbf{C}_{cb_{\beta_u}}\right) d\mathbf{b}_{\beta_u} \tag{B.35}$$

but the unbounded integral is equal to unity, so

$$\propto p(\boldsymbol{\beta}_u) \, \Upsilon_{cb_{\beta_u}} \tag{B.36}$$

When sampling $\boldsymbol{\beta}_u$, those parts of $\Upsilon_{cb_{\beta_u}}$ which are independent of $\boldsymbol{\beta}_u$ can be neglected, resulting in equation (5.32).

B.3   Indicators for sinusoidal basis functions

In §7.3.4.1, it is necessary to sample from the distribution

$$p(\boldsymbol{\gamma}_u \mid \boldsymbol{\gamma}_f, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$$

$$\propto p(\boldsymbol{\gamma}_u) \int \underbrace{p(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_\gamma, k, \mathbf{a}^{(k)}, \sigma_e^2)}_{\text{Likelihood}} \, p(\mathbf{c}_{\gamma_u} \mid \sigma_c^2) \, d\mathbf{c}_{\gamma_u} \tag{B.37}$$

From the modelling equations for the sinusoid + AR model (eq. 7.20, 7.21 & 7.23), we have

$$\mathbf{e} = \mathbf{A}^{(k)}\mathbf{w} = \mathbf{A}^{(k)}\mathbf{x} - \mathbf{A}^{(k)}\mathbf{s} = \mathbf{A}^{(k)}\mathbf{x} - \mathbf{A}^{(k)}\mathbf{G}\left(\mathbf{c} \circ \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\gamma} \end{bmatrix}\right) \tag{B.38}$$

where $\circ$ denotes the elementwise product, and the basis functions are arranged such that the in-phase and quadrature bases appear in positions $\{1 \ldots n_\gamma\}$ and $\{n_\gamma + 1 \ldots 2n_\gamma\}$ respectively, in the same order, where $n_\gamma = \frac{1}{2}n_c$.

Hence the likelihood (eq. 7.27) can be rewritten as

$$p\big(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}}, k, \mathbf{a}^{(k)}, \sigma_e^2\big) = p_e\big(\mathbf{A}^{(k)}\mathbf{x} - \mathbf{A}^{(k)}\mathbf{G}_{\gamma_f}\mathbf{c}_{\gamma_f} - \mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}\mathbf{c}_{\gamma_u} \mid \sigma_e^2\big) \quad \text{(B.39)}$$

$$= \mathbf{N}\big(\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}\mathbf{c}_{\gamma_u} \mid \underbrace{\mathbf{A}^{(k)}(\mathbf{x} - \mathbf{G}_{\gamma_f}\mathbf{c}_{\gamma_f})}_{\triangleq \, \mathbf{e}_f}, \sigma_e^2\mathbf{I}_{n_e}\big) \quad \text{(B.40)}$$

where $\mathbf{e}_f$ is the excitation signal which would need to be applied to the AR model to produce $\mathbf{x}$ if only the sinusoids in the partition $(\cdot)_{\gamma_f}$ were used.

This Gaussian can then be rearranged, using the identity of §A.2, as

$$p\big(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}}, k, \mathbf{a}^{(k)}, \sigma_e^2\big) = \Upsilon_{lc_{\gamma_u}} \times \mathbf{N}\big(\mathbf{c}_{\gamma_u} \mid \boldsymbol{\mu}_{lc_{\gamma_u}}, \mathbf{C}_{lc_{\gamma_u}}\big) \quad \text{(B.41)}$$

where

$$\mathbf{C}_{lc_{\gamma_u}} = \sigma_e^2 \left(\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}\right)^{-1} \quad \text{(B.42)}$$

$$\boldsymbol{\mu}_{lc_u} = \left(\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}\right)^{-1}\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{e}_f \quad \text{(B.43)}$$

$$\Upsilon_{lc_u} = \frac{\exp\big(-\frac{1}{2\sigma_e^2}\big[\mathbf{e}_f^T\mathbf{e}_f - \mathbf{e}_f^T\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}(\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u})^{-1}\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{e}_f\big]\big)}{(\sqrt{2\pi}\sigma_e)^{n_e - n_{c_{\gamma_u}}} \sqrt{\big|\mathbf{G}_{\gamma_u}^T\mathbf{A}^{(k)T}\mathbf{A}^{(k)}\mathbf{G}_{\gamma_u}\big|}} \quad \text{(B.44)}$$

where $n_{c_{\gamma_u}}$ is the length of the vector $\mathbf{c}_{\gamma_u}$, which can either be zero or two, as there is only one indicator in γ_u, and it controls a pair of basis functions (see §7.3.1.2).

Now that the likelihood is in the convenient form of a Gaussian in $\mathbf{c}_{\gamma_u}$, it can be multiplied by the prior on $\mathbf{c}_{\gamma_u}$ (eq. 7.28) to form the integrand of equation (B.37):

$$p(\mathbf{x}, \mathbf{c}_{\gamma_u} \mid \boldsymbol{\gamma}, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$$

$$= p(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_{\boldsymbol{\gamma}}, k, \mathbf{a}^{(k)}, \sigma_e^2)\, p(\mathbf{c}_{\gamma_u} \mid \sigma_c^2) \quad \text{(B.45)}$$

$$= \Upsilon_{lc_{\gamma_u}} \mathbf{N}\big(\mathbf{c}_{\gamma_u} \mid \boldsymbol{\mu}_{lc_{\gamma_u}}, \mathbf{C}_{lc_{\gamma_u}}\big) \mathbf{N}\big(\mathbf{c}_{\gamma_u} \mid \mathbf{0}, \sigma_c^2\mathbf{I}_{n_{c_{\gamma_u}}}\big) \quad \text{(B.46)}$$

which is a product of Gaussians and so can be simplified using the identity of §A.1 to

$$= \Upsilon_{cc_{\gamma u}} \times \mathbf{N}\big(\mathbf{c}_{\gamma u} \mid \boldsymbol{\mu}_{cc_{\gamma u}}, \mathbf{C}_{cc_{\gamma u}}\big) \tag{B.47}$$

where

$$\mathbf{C}_{cc_{\gamma u}} = \big(\mathbf{C}_{lc_{\gamma u}}^{-1} + \tfrac{1}{\sigma_c^2}\mathbf{I}_{n_{c_{\gamma u}}}\big)^{-1} \tag{B.48}$$

$$\boldsymbol{\mu}_{cc_{\gamma u}} = \mathbf{C}_{cc_{\gamma u}} \mathbf{C}_{lc_{\gamma u}}^{-1} \boldsymbol{\mu}_{lc_{\gamma u}} \tag{B.49}$$

$$\Upsilon_{cc_{\gamma u}} = \frac{\Upsilon_{lc_{\gamma u}}}{\sqrt{2\pi}^{n_{c_{\gamma u}}}} \sqrt{\frac{|\mathbf{C}_{cc_{\gamma u}}|}{|\mathbf{C}_{lc_{\gamma u}}| |\sigma_c^2 \mathbf{I}_{n_{c_{\gamma u}}}|}} \exp\big(-\tfrac{1}{2}\big[\boldsymbol{\mu}_{lc_{\gamma u}}^T \mathbf{C}_{lc_{\gamma u}}^{-1} \boldsymbol{\mu}_{lc_{\gamma u}} - \boldsymbol{\mu}_{cc_{\gamma u}}^T \mathbf{C}_{cc_{\gamma u}}^{-1} \boldsymbol{\mu}_{cc_{\gamma u}}\big]\big) \tag{B.50}$$

which simplify, after much cancellation, to

$$\mathbf{C}_{cc_{\gamma u}} = \big(\tfrac{1}{\sigma_e^2}\mathbf{G}_{\gamma u}^T \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \mathbf{G}_{\gamma u} + \tfrac{1}{\sigma_c^2}\mathbf{I}_{n_{c_{\gamma u}}}\big)^{-1} \tag{B.51}$$

$$\boldsymbol{\mu}_{cc_{\gamma u}} = \tfrac{1}{\sigma_e^2}\mathbf{C}_{cc_{\gamma u}} \mathbf{G}_{\gamma u}^T \mathbf{A}^{(k)T} \mathbf{e}_f \tag{B.52}$$

$$\Upsilon_{cc_{\gamma u}} = \frac{\sqrt{|\mathbf{C}_{cc_{\gamma u}}|}}{\big(\sqrt{2\pi}\sigma_e\big)^{n_e} \sigma_c^{n_{c_{\gamma u}}}} \exp\Big(-\tfrac{1}{2\sigma_e^2}\big[\mathbf{e}_f^T \mathbf{e}_f - \tfrac{1}{\sigma_e^2}\mathbf{e}_f^T \mathbf{A}^{(k)} \mathbf{G}_{\gamma u} \mathbf{C}_{cc_{\gamma u}} \mathbf{G}_{\gamma u}^T \mathbf{A}^{(k)T} \mathbf{e}_f\big]\Big) \tag{B.53}$$

Now equation (B.37) can be written in the form

$$p(\boldsymbol{\gamma}_u \mid \boldsymbol{\gamma}_f, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)$$

$$\propto p(\boldsymbol{\gamma}_u)\Upsilon_{cc_{\gamma u}} \int \mathbf{N}\big(\mathbf{c}_{\gamma u} \mid \boldsymbol{\mu}_{cc_{\gamma u}}, \mathbf{C}_{cc_{\gamma u}}\big)\, d\mathbf{c}_{\gamma u} \tag{B.54}$$

but the unbounded integral is equal to unity, so

$$\propto p(\boldsymbol{\gamma}_u)\Upsilon_{cc_{\gamma u}} \tag{B.55}$$

Sampling step (7.32) requires the full conditional distribution for $\mathbf{c}_{\gamma u}$. This can be obtained from the likelihood using Bayes' theorem (eq. 2.3):

$$p(\mathbf{c}_{\gamma u} \mid \boldsymbol{\gamma}, \mathbf{c}_{\gamma_f}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2) \propto p(\mathbf{x} \mid \boldsymbol{\gamma}, \mathbf{c}_{\gamma}, k, \mathbf{a}^{(k)}, \sigma_c^2, \sigma_e^2)\, p(\mathbf{c}_{\gamma u} \mid \sigma_c^2) \tag{B.56}$$

which has already been derived (eq. B.47) as

$$\propto \mathbf{N}\left(\mathbf{c}_{\gamma_u} \mid \boldsymbol{\mu}_{c\mathbf{c}_{\gamma_u}}, \mathbf{C}_{c\mathbf{c}_{\gamma_u}}\right) \tag{B.57}$$

Demonstration CD

C

This thesis is accompanied by a compact disc containing examples of signals processed by the restoration algorithms. All signals were processed at 44.1 kHz, and have had TPDF dither (see §7.1.4.1) added before being quantised to 16 bits.

The vocal extract **winner** is taken from the left channel of the track *I Am What You See* on the album *Last Dance* by Jason Rebello and Joy Rose.[1] The extract **violins** is taken from the right channel of the track *Violins* on the CD *Sound Check* by Alan Parsons and Stephen Court. The extract **piano** is taken from *Une Larme* by Mussorgsky, performed by Jenő Jandó,[2] using the left channel.

---

[1] Catalogue number ATJR001.
[2] Available on the Naxos CD 8.550044.

Table C.1. Tracks on the accompanying demonstration CD.

Track	Page	Source signal	Algorithm	Relevant parameters	Output
1	(p. 66)	**winner**	Added Gaussian noise	$\sigma_\nu = 100$	—
2	(p. 66)	**winner**	—	—	—
3	(p. 67)	Track 2	Simulation smoother	$k = 30$	mean x
4	(p. 67)	Track 2	Simulation smoother	k variable	mean x
5	(p. 67)	Track 2	Spectral subtraction	—	—
6	(p. 108)	**violins**	—	—	—
7	(p. 109)	**violins**	NAR distorted	5 third degree terms, up to lag 9	—
8	(p. 110)	Track 7	AR-NAR restoration	220 candidate third degree terms up to lag 10	mean x
9	(p. 116)	**piano**	—	—	—
10	(p. 116)	**piano**	Quantised	$\Delta = 50$	—
11	(p. 116)	Track 10	Quantisation error, amplified	—	—
12	(p. 129)	Track 10	AR restoration	k variable	mean x
13	(p. 131)	Track 10	AR restoration	k variable	final x
14	(p. 131)	Track 10	AR restoration	$k = 40$	mean x
15	(p. 131)	Track 10	AR restoration	$k = 2$	mean x
16	(p. 139)	Track 10	AR + sinusoids restoration	k, n_ζ variable	mean x
17	(p. 141)	Track 10	AR + sinusoids restoration	k variable, $n_\gamma = 30$	mean x

Bibliography

[1] *AES17-1998: AES standard method for digital audio engineering — Measurement of digital audio equipment.* Audio Engineering Society, 1998.

[2] H. Akaike. "A new look at statistical model identification". *IEEE Transactions on Automatic Control*, **AC-19**, pp. 716–723, 1974.

[3] H. Akaike. "The interpretation of improper prior distributions as limits of data dependent proper prior distributions". *Journal of the Royal Statistical Society B*, **42**, pp. 46–52, 1980.

[4] J. Aldred. *Manual of Sound Recording.* Fountain Press, Argus Books, 1978.

[5] J. Allen. "Short term spectral analysis, synthesis and modification". *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-25**, pp. 235–239, 1977.

[6] Y. Amit. "On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions". *Journal of Multivariate Analysis*, **38**, pp. 82–99, 1991.

[7] C. Andrieu & A. Doucet. "Joint Bayesian detection and estimation of noisy sinusoids via reversible jump MCMC". *IEEE Transactions on Signal Processing*, 1999. To appear.

[8] C. Andrieu, A. Doucet & P. Duvaut. *Fully Bayesian Joint Estimation of the Dimension and Parameters of an AR Model Using MCMC.* Tech. rep., Equipe Traitement des Images et du Signal, EN-SEA, France, 1997.

[9] Anonymous. "Getting the goat". *The Economist*, **350**(8107), p. 102, 1999.

[10] M. M. Barbieri & A. O'Hagan. *A reversible jump MCMC sampler for Bayesian analysis of ARMA time series.* Tech. rep., Dipartimento di Statistica, Probabilità e Statistiche Applicate, Università "La Sapienza", Roma and Department of Mathematics, University of Nottingham, 1996.

[11] D. A. Barlow & G. R. Garside. "Groove deformation and distortion in records". *Journal of the Audio Engineering Society*, **26**(7–8), pp. 498–510, 1978.

[12] G. Barnett, R. Kohn & S. Sheather. *Bayesian Estimation of an Autoregressive Model using Markov Chain Monte Carlo*. Tech. rep., Australian Graduate School of Management, University of New South Wales, 1994.

[13] G. Barnett, R. Kohn & S. Sheather. "Bayesian estimation of an autoregressive model using Markov chain Monte Carlo". *Journal of Econometrics*, **74**(2), pp. 237–254, 1996.

[14] A. Barron, J. Rissanen & B. Yu. "The minimum description length principle in coding and modeling". *IEEE Transactions on Information Theory*, **44**(6), pp. 2743–2760, 1998.

[15] T. Bayes. "An essay toward solving a problem in the doctrine of chances". *Philosophical Transactions of the Royal Society of London*, **53**, pp. 370–418, 1763.

[16] J. O. Berger & L. R. Pericchi. "The intrinsic Bayes factor for model selection and prediction". *Journal of the American Statistical Association*, **91**(433), pp. 109–122, 1996.

[17] J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith, eds. *Bayesian Statistics 4*. Oxford University Press, 1992.

[18] J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith, eds. *Bayesian Statistics 5*. Oxford University Press, 1996.

[19] J. M. Bernardo & A. F. M. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.

[20] J. Besag. "A candidate's formula — a curious result in Bayesian prediction". *Biometrika*, **76**(1), p. 183, 1989.

[21] S. A. Billings, S. Chen & M. J. Korenberg. "Identification of MIMO non-linear systems using a forward-regression orthogonal estimator". *International Journal of Control*, **49**(6), pp. 2157–2189, 1989.

[22] W. B. Bishop & P. M. Djurić. "Model order selection of damped sinusoids in noise by predictive densities". *IEEE Transactions on Signal Processing*, **44**(3), pp. 611–619, 1996.

[23] G. E. P. Box, G. M. Jenkins & G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Holden-Day, 3rd edn., 1994.

[24] G. E. P. Box & G. C. Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, 1973.

[25] P. M. T. Broersen & H. E. Wensink. "On the penalty factor for autoregressive order selection in finite samples". *IEEE Transactions on Signal Processing*, **44**(3), pp. 748–752, 1996.

[26] S. P. Brooks. "Markov chain Monte Carlo and its application". *The Statistician*, **47**(1), pp. 69–100, 1998.

[27] O. Cappé & J. Laroche. "Evaluation of short-time spectral attenuation techniques for the restoration of musical recordings". *IEEE Transactions on Speech and Audio Processing*, **3**(1), pp. 84–93, 1995.

[28] B. P. Carlin & S. Chib. "Bayesian model choice via Markov chain Monte Carlo". *Journal of the Royal Statistical Society B*, **57**, pp. 473–484, 1995.

[29] B. P. Carlin, N. G. Polson & D. S. Stoffer. "A Monte Carlo approach to nonnormal and nonlinear state-space modeling". *Journal of the American Statistical Association*, **87**(418), pp. 493–500, 1992.

[30] C. K. Carter & R. Kohn. "Markov-chain Monte-Carlo in conditionally Gaussian state-space models". *Biometrika*, **83**(3), pp. 589–601, 1996.

[31] J. E. Chance, K. Worden & G. R. Tomlinson. "Frequency domain analysis of NARX neural networks". *Journal of Sound and Vibration*, **213**(5), pp. 915–941, 1998.

[32] C. W. S. Chen. "Bayesian analysis of bilinear time series models: A Gibbs sampling approach". *Communications in Statistics—Theory and Methods*, **21**(12), pp. 3407–3425, 1992.

[33] C. W. S. Chen. *On the Selection of Best Subset Autoregressive Time Series Models*. Tech. rep., Department of Statistics, Feng-Chia University, Taiwan, 1996.

[34] R. Chen, J. S. Liu & R. S. Tsay. "Additivity tests for nonlinear autoregression". *Biometrika*, **82**(2), pp. 369–383, 1995.

[35] R. Chen & R. S. Tsay. "Functional-coefficient autoregressive models". *Journal of the American Statistical Association*, **88**(421), pp. 298–308, 1993.

[36] R. Chen & R. S. Tsay. "Nonlinear additive ARX models". *Journal of the American Statistical Association*, **88**(423), pp. 955–967, 1993.

[37] S. Chen & S. A. Billings. "Modelling and analysis of non-linear time series". *International Journal of Control*, **50**(6), pp. 2151–2171, 1989.

[38] S. Chen & S. A. Billings. "Representations of non-linear systems: The NARMAX model". *International Journal of Control*, **49**(3), pp. 1013–1032, 1989.

[39] S. Chen, S. A. Billings & W. Luo. "Orthogonal least squares methods and their application to non-linear system identification". *International Journal of Control*, **50**(5), pp. 1873–1896, 1989.

[40] S. Chib. *Bayesian Estimation and Comparison of Multiple Change Point Models*. Tech. rep., John M. Olin School of Business, Washington University, 1996.

[41] C. Colomes, M. Lever, J. B. Rault, Y. F. Dehery & G. Faucon. "A perceptual model applied to audio bit-rate reduction". *Journal of the Audio Engineering Society*, **43**(4), pp. 233–240, 1995.

[42] D. H. Cooper. "Compensation for tracing and tracking error". *Journal of the Audio Engineering Society*, **11**(4), pp. 659–662, 1963.

[43] M. S. Corrington. "Tracing distortion in phonograph records". *RCA Review*, **10**(12), pp. 241–253, 1949.

[44] P. Craven & M. Gerzon. "Lossless coding for audio discs". *Journal of the Audio Engineering Society*, **44**(9), pp. 706–720, 1996.

[45] A. Czyzewski. "Learning algorithms for audio signal enhancement: (1) neural network implementation for the removal of impulse distortions". *Journal of the Audio Engineering Society*, **45**(10), pp. 815–831, 1997.

[46] P. de Jong & N. Shephard. "The simulation smoother for time series models". *Biometrika*, **82**(2), pp. 339–350, 1995.

[47] P. Dellaportas, J. J. Forster & I. Ntzoufras. *On Bayesian Model and Variable Selection Using MCMC*. Tech. rep., Department of Statistics, Athens University of Economics and Business, 1997.

[48] J. R. Deller, J. G. Proakis & J. H. L. Hansen. *Discrete-time processing of speech signals*. Macmillan, 1993.

[49] B. Denckla. "Subtractive dither for internet audio". *Journal of the Audio Engineering Society*, **46**(7–8), pp. 654–656, 1998.

[50] J. R. Dickie & A. K. Nandi. "A comparative study of AR order selection methods". *Signal Processing*, **40**, pp. 239–255, 1994.

[51] P. M. Djurić, S. J. Godsill, W. J. Fitzgerald & P. J. W. Rayner. "Detection and estimation of signals by reversible jump Markov chain Monte Carlo computations". *Proceedings of IEEE ICASSP-98*, **4**, pp. 2269–2272, 1998.

[52] P. M. Djurić & S. M. Kay. "Order selection of autoregressive models". *IEEE Transactions on Signal Processing*, **40**(11), pp. 2829–2833, 1992.

[53] A. Eiling & W. Schmitt. "A comprehensive picture of magnetic recording: Theory and experiment; part II: Analog recording (audio and video)". *Journal of Magnetism and Magnetic Materials*, **130**, pp. 416–432, 1994.

[54] Y. Ephraim. "Statistical-model-based speech enhancement systems". *Proceedings of the IEEE*, **80**(10), pp. 1526–1555, 1992.

[55] S. Y. Fakhouri. "Identification of Volterra kernels of nonlinear systems". *Proceedings of the IEE, Part D*, **127**(6), pp. 296–304, 1980.

[56] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2nd edn., 1987.

[57] S. Frühwirth-Schnatter. "Data augmentation and dynamic linear models". *Journal of Time Series Analysis*, **15**, pp. 183–202, 1994.

[58] G. M. Furnival & R. W. Wilson, Jr. "Regressions by leaps and bounds". *Technometrics*, **16**(4), pp. 499–511, 1974.

[59] A. E. Gelfand, D. K. Dey & H. Chang. "Model determination using predictive distributions with implementation via sampling-based methods". In [17], pp. 147–167.

[60] A. E. Gelfand, S. K. Sahu & B. P. Carlin. "Efficient parametrizations for generalised linear mixed models". In [18], pp. 165–180. With discussion.

[61] A. E. Gelfand & A. F. M. Smith. "Sampling-based approaches to calculating marginal densities". *Journal of the American Statistical Association*, **85**(410), pp. 398–409, 1990.

[62] A. Gelman, J. B. Carlin, H. S. Stern & D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall, 1995.

[63] A. Gelman & D. B. Rubin. "Inference from iterative simulation using multiple sequences". *Statistical Science*, **7**(4), pp. 457–511, 1992. With discussion.

[64] A. Gelman & D. B. Rubin. "A single series from the Gibbs sampler provides a false sense of security". In [17], pp. 625–631.

[65] S. Geman & D. Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(6), pp. 721–741, 1984.

[66] E. I. George & R. E. McCulloch. "Variable selection via Gibbs sampling". *Journal of the American Statistical Association*, **88**(423), pp. 881–889, 1993.

[67] E. I. George & R. E. McCulloch. "Stochastic search variable selection". In [75], pp. 203–214.

[68] E. I. George & R. E. McCulloch. "Approaches for Bayesian variable selection". *Statistica Sinica*, **7**(2), pp. 339–373, 1997.

[69] M. A. Gerzon & P. G. Craven. "A high-rate buried-data channel for audio CD". *Journal of the Audio Engineering Society*, **43**(1–2), pp. 3–22, 1995.

[70] J. Geweke. *Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints and the Evaluation of Constraint Probabilities*. Tech. rep., Department of Economics, University of Minnesota, 1992.

[71] J. Geweke. "Variable selection and model comparison in regression". In [18], pp. 609–620.

[72] C. J. Geyer. "Practical Markov chain Monte Carlo". *Statistical Science*, **7**(4), pp. 473–511, 1992. With discussion.

[73] C. J. Geyer. "Burn-in is unnecessary". Available from http://www.stat.umn.edu/~charlie/mcmc/burn.html, 1998.

[74] C. J. Geyer & E. A. Thompson. "Annealing Markov-chain Monte-Carlo with applications to ancestral analysis". *Journal of the American Statistical Association*, **90**(431), pp. 909–920, 1995.

[75] W. R. Gilks, S. Richardson & D. J. Spiegelhalter, eds. *Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics*. Chapman & Hall, 1996.

[76] W. R. Gilks & G. O. Roberts. "Strategies for improving MCMC". In [75], pp. 89–114.

[77] W. R. Gilks, G. O. Roberts & S. K. Sahu. *Adaptive Markov Chain Monte Carlo*. Tech. rep., Medical Research Council Biostatistics Unit, Cambridge and Statistical Laboratory, University of Cambridge, 1996.

[78] S. J. Godsill. *The Restoration of Degraded Audio Signals*. Ph.D. thesis, University of Cambridge, 1993.

[79] S. J. Godsill. "Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes". *International Statistical Review*, **65**(1), pp. 1–21, 1997.

[80] S. J. Godsill. "Robust modelling of noisy ARMA signals". *Proceedings of IEEE ICASSP-97*, **V**, pp. 3797–3800, 1997.

[81] S. J. Godsill. *Some new relationships between MCMC model uncertainty methods*. Tech. Rep. CUED/F-INFENG/TR.305, Department of Engineering, University of Cambridge, 1997.

[82] S. J. Godsill & P. J. W. Rayner. "A Bayesian approach to the detection and correction of bursts of errors in audio signals". *Proceedings of IEEE ICASSP-92*, **II**, pp. 261–264, 1992.

[83] S. J. Godsill & P. J. W. Rayner. "A Bayesian approach to the restoration of degraded audio signals". *IEEE Transactions on Speech and Audio Processing*, **3**(4), pp. 267–278, 1995.

[84] S. J. Godsill & P. J. W. Rayner. "Robust noise modelling with application to audio restoration". In [136].

[85] S. J. Godsill & P. J. W. Rayner. "Robust noise reduction for speech and audio signals". *Proceedings of IEEE ICASSP-96*, **II**, pp. 625–628, 1996.

[86] S. J. Godsill & P. J. W. Rayner. "Robust treatment of impulsive noise in speech and audio signals". In J. O. Berger, B. Betro, E. Moreno, L. R. Pericchi, F. Ruggeri, G. Salinetti & L. Wasserman, eds., *Bayesian Robustness*, vol. 29, pp. 331–342. IMS Lecture Notes – Monograph Series, 1996.

[87] S. J. Godsill & P. J. W. Rayner. *Digital Audio Restoration: A Statistical Model Based Approach*. Springer-Verlag, 1998.

[88] S. J. Godsill & P. J. W. Rayner. "Statistical reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler". *IEEE Transactions on Speech and Audio Processing*, **6**(4), pp. 352–372, 1998.

[89] S. J. Godsill, P. J. W. Rayner & O. Cappé. "Digital audio restoration". In K. Brandenburg & M. Kahrs, eds., *Applications of Digital Signal Processing to Audio and Acoustics*, chap. 4, pp. 133–194. Kluwer Academic Publishers, 1996.

[90] Z. Goh, K.-C. Tan & B. T. G. Tan. "Speech enhancement based on a voiced-unvoiced speech model". *Proceedings of IEEE ICASSP-98*, **1**, pp. 401–404, 1998.

[91] W. Greblicki & M. Pawlak. "Cascade non-linear system identification by a non-parametric method". *International Journal of Systems Science*, **25**(1), pp. 129–153, 1994.

[92] P. J. Green. "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". *Biometrika*, **82**(4), pp. 711–732, 1995.

[93] U. Grenander & M. I. Miller. "Representations of knowledge in complex systems". *Journal of the Royal Statistical Society B*, **56**(4), pp. 549–603, 1994.

[94] F. Gustafsson & H. Hjalmarsson. "Twenty-one ML estimators for model selection". *Automatica*, **31**(10), pp. 1377–1392, 1995.

[95] R. Haber & H. Unbehauen. "Structure identification of nonlinear dynamic systems—a survey on input/output approaches". *Automatica*, **26**(4), pp. 651–677, 1990.

[96] V. A. Hajivassiliou & P. A. Ruud. "Classical estimation methods for LDV models using simulation". In R. F. Engle & D. L. McFadden, eds., *Handbook of Econometrics*, vol. 4, pp. 2383–2441. North-Holland, 1994.

[97] J. M. Hammersley & D. C. Handscomb. *Monte Carlo Methods*. Methuen's Monographs on Applied Probability and Statistics. Methuen & Co, 1964.

[98] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". *Biometrika*, **57**(1), pp. 97–109, 1970.

[99] D. Haughton. "Consistency of a class of information criteria for model selection in nonlinear regression". *Theory of Probability and its Applications*, **37**(1), pp. 47–53, 1992.

[100] J. Heitkötter & D. Beasley, eds. *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)*. USENET: comp.ai.genetic[1], 1998.

[101] R. R. Hocking & R. N. Leslie. "Best subset in regression analysis". *Technometrics*, **9**, pp. 431–540, 1967.

[102] J. H. Holland. *Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[103] J. L. Hood. "Tape recording". In I. R. Sinclair, ed., *Audio Electronics Reference Book*, pp. 155–205. BSP Professional Books, 1989.

---

[1]Also available by anonymous FTP from:
   ftp://rtfm.mit.edu/pub/usenet/news.answers/ai-faq/genetic/

[104] G. Huerta & M. West. *Priors and Component Structures in Autoregressive Time Series Models*. Tech. rep., Institute of Statistics & Decision Sciences, Duke University, 1997.

[105] B. Jansson. *Random Number Generators*. Victor Pettersons Bokindustri Aktiebolag, Stockholm, 1966.

[106] E. T. Jaynes. "Probability theory: The logic of science". Available from ftp://bayes.wustl.edu/pub/Jaynes/book.probability.theory/, 1994.

[107] H. Jeffreys. *Theory of Probability*. Oxford University Press, 3rd edn., 1961.

[108] N. L. Johnson & S. Kotz. *Distributions in Statistics: Continuous Univariate Distributions*. Wiley, 1970.

[109] R. H. Jones. "Fitting a continuous time autoregression to discrete data". In D. F. Findley, ed., *Applied Time Series Analysis II*, pp. 651–682. Academic Press, 1981.

[110] K. Judd & A. Mees. "On selecting models for nonlinear time series". *Physica D*, **82**, pp. 426–444, 1995.

[111] T. Kasparis & J. Lane. "Adaptive scratch noise filtering". *IEEE Transactions on Consumer Electronics*, **39**(4), pp. 917–922, 1993.

[112] R. E. Kass & A. E. Raftery. "Bayes factors". *Journal of the American Statistical Association*, **90**(430), pp. 773–795, 1995.

[113] S. M. Kay & V. Nagesha. "Maximum likelihood estimation of signals in autoregressive noise". *IEEE Transactions on Signal Processing*, **42**(1), pp. 88–101, 1994.

[114] G. Kitagawa & W. Gersch. *Smoothness Priors Analysis of Time Series*. Springer-Verlag, 1996.

[115] W. Klippel. "Adaptive adjustment of nonlinear filters used for loudspeaker equalization". Presented at the *104th Convention of the Audio Engineering Society*, preprint 4646, 1998.

[116] E. S. Koneva. "Model fitting for real-data time series". *Automation & Remote Control*, **49**, pp. 691–702, 1988.

[117] J. H. Kotecha & P. M. Djurić. "Gibbs sampling approach for generation of multivariate Gaussian random variables". *Proceedings of IEEE ICASSP-99*, **3**, pp. 1757–1760, 1999.

[118] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 6th edn., 1988.

[119] L. Kuo & B. Mallick. *Variable Selection for Regression Models*. Tech. Rep. 94-26, Department of Statistics, University of Connecticut, 1994.

[120] L. Kuo & B. Mallick. "Bayesian semiparametric inference for the accelerated failure-time model". *Canadian Journal of Statistics*, **25**(4), pp. 457–472, 1997.

[121] L. Kuo & B. Mallick. "Variable selection for regression models". *Sankhyā B*, 1998. To appear.

[122] J. Laroche. "Removing preechos from audio recordings". In [136].

[123] W. D. Lewis & F. V. Hunt. "A theory of tracing distortion in sound reproduction in phonograph records". *Journal of the Acoustical Society of America*, **12**(3), pp. 348–365, 1941.

[124] J. S. Lim & A. V. Oppenheim. "All-pole modelling of degraded speech". *IEEE Transactions on Acoustics, Speech and Signal Processing*, **ASSP-26**(3), pp. 197–210, 1978.

[125] J. S. Lim & A. V. Oppenheim. "Enhancement and bandwidth compression of noisy speech". *Proceedings of the IEEE*, **67**(12), pp. 1586–1604, 1979.

[126] D. V. Lindley. "A statistical paradox". *Biometrika*, **44**, pp. 187–192, 1957.

[127] S. P. Lipshitz, R. A. Wannamaker & J. Vanderkooy. "Quantization and dither: a theoretical survey". *Journal of the Audio Engineering Society*, **40**(5), pp. 355–375, 1992.

[128] G. P. Liu, V. Kadirkamanathan & S. A. Billings. "On-line identification of nonlinear systems using Volterra polynomial basis function neural networks". *Neural Networks*, **11**(9), pp. 1645–1657, 1998.

[129] J. S. Liu & R. Chen. "Sequential Monte Carlo methods for dynamic systems". *Journal of the American Statistical Association*, **93**(443), pp. 1032–1044, 1998.

[130] J. S. Liu, W. H. Wong & A. Kong. "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes". *Biometrika*, **81**(1), pp. 27–40, 1994.

[131] J. S. Liu, W. H. Wong & A. Kong. "Covariance structure and convergence rate of the Gibbs sampler with various scans". *Journal of the Royal Statistical Society B*, **57**(1), pp. 157–169, 1995.

[132] G. M. Ljung & G. E. P. Box. "The likelihood function of stationary autoregressive-moving average models". *Biometrika*, **66**(2), pp. 265–270, 1979.

[133] W. Luo & S. A. Billings. "Adaptive model selection and estimation for nonlinear systems using a sliding data window". *Signal Processing*, **46**, pp. 179–202, 1995.

[134] D. Madigan & J. York. "Bayesian graphical models for discrete data". *International Statistical Review*, **63**(2), pp. 215–232, 1995.

[135] R. C. Maher. "On the nature of granulation noise in uniform quantization systems". *Journal of the Audio Engineering Society*, **40**(1–2), pp. 12–20, 1992.

[136] R. C. Maher, ed. *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*. 1995.

[137] J. Marriott, N. Ravishanker, A. Gelfand & J. Pai. "Bayesian analysis of ARMA processes: Complete sampling-based inference under exact likelihoods". In D. A. Berry, K. M. Chaloner & J. K. Geweke, eds., *Bayesian Analysis in Statistics and Econometrics*, pp. 243–256. John Wiley & Sons, 1996.

[138] G. Marsaglia. "Generating a variable from the tail of the normal distribution". *Technometrics*, **6**(1), pp. 101–102, 1964.

[139] R. E. McCulloch & R. S. Tsay. "Bayesian analysis of autoregressive time series via the Gibbs sampler". *Journal of Time Series Analysis*, **15**(2), pp. 235–250, 1994.

[140] K. L. Mengersen, C. P. Robert & C. Guihenneuc-Jouyaux. "MCMC convergence diagnostics: a revieWWW". In J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith, eds., *Bayesian Statistics 6*. Oxford University Press, 1999. To appear.

[141] K. J. Mercer. *Identification of Distortion Models*. Ph.D. thesis, University of Cambridge, 1993.

[142] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller & E. Teller. "Equation of state calculations by fast computing machines". *Journal of Chemical Physics*, **21**(6), pp. 1087–1092, 1953.

[143] I. J. Morrison. *The Application of Volterra Series to Signal Detection and Estimation*. Ph.D. thesis, University of Cambridge, 1990.

[144] R. M. Neal. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Tech. Rep. CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

[145] H. Nyquist. "Certain topics in telegraph transmission theory". *Transactions of the American Institute of Electrical Engineers*, **47**, pp. 617–644, 1928.

[146] J. J. K. Ó Ruanaidh & W. J. Fitzgerald. "Interpolation of missing samples for audio restoration". *Electronics Letters*, **30**, pp. 622–623, 1994.

[147] J. J. K. Ó Ruanaidh & W. J. Fitzgerald. *The restoration of audio gramophone recordings using Gibbs sampling*. Tech. Rep. CUED/F-INFENG/TR.153, Department of Engineering, University of Cambridge, 1994.

[148] J. J. K. Ó Ruanaidh & W. J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing*. Statistics and Computing. Springer, 1996.

[149] A. O'Hagan. "Fractional Bayes factors for model comparison". *Journal of the Royal Statistical Society B*, **57**(1), pp. 99–138, 1995.

[150] N. G. Polson. "Convergence of Markov chain Monte Carlo algorithms". In [18], pp. 297–321. With discussion.

[151] K. J. Pope. *Time Series Analysis*. Ph.D. thesis, University of Cambridge, 1993.

[152] K. J. Pope & P. J. W. Rayner. "Non-linear system identification using Bayesian inference". *Proceedings of IEEE ICASSP-94*, **IV**, pp. 457–460, 1994.

[153] R. Prado & M. West. *Exploratory Modelling of Multiple Non-Stationary Time Series: Latent Process Structure and Decompositions*. Tech. rep., Institute of Statistics & Decision Sciences, Duke University, 1997.

[154] D. Preis & H. Polchlopek. "Restoration of nonlinearly distorted magnetic recordings". *Journal of the Audio Engineering Society*, **32**(1), pp. 26–30, 1984.

[155] W. H. Press, S. A. Teukolsky, W. T. Vetterling & B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edn., 1992.

[156] A. E. Raftery, D. Madigan & J. A. Hoeting. "Bayesian model averaging for linear regression models". *Journal of the American Statistical Association*, **92**(437), pp. 179–191, 1987.

[157] J. J. Rajan & P. J. W. Rayner. "Unsupervised time-series classification". *Signal Processing*, **46**(1), pp. 57–74, 1995.

[158] P. J. W. Rayner & S. J. Godsill. "The detection and correction of artefacts in archived gramophone recordings". In *Proceedings of the IEEE Workshop on Audio and Acoustics*. Mohonk, NY State, 1991.

[159] H. Redlich & H.-J. Klemp. "A new method of disc recording for reproduction with reduced distortion: the tracing simulator". *Journal of the Audio Engineering Society*, **13**(2), pp. 111–118, 1965.

[160] M. J. Reed & M. O. Hawksford. "Practical modelling of nonlinear audio systems using the Volterra series". Presented at the *100th Convention of the Audio Engineering Society*, preprint 4264, 1996.

[161] M. J. Reed & M. O. J. Hawksford. "Comparison of audio system nonlinear performance in Volterra space". Presented at the *103rd Convention of the Audio Engineering Society*, preprint 4606, 1997.

[162] C. P. Robert. "Simulation of truncated normal variables". *Statistics and Computing*, **5**, pp. 121–125, 1995.

[163] G. O. Roberts & S. K. Sahu. "Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler". *Journal of the Royal Statistical Society B*, **59**(2), pp. 291–317, 1997.

[164] L. G. Roberts. "Picture coding using pseudo-random noise". *IRE Transactions on Information Theory*, **IT-8**, pp. 145–154, 1962.

[165] G. M. K. Saleh. *Bayesian Inference in Speech Processing*. Ph.D. thesis, University of Cambridge, 1996.

[166] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley & Sons, 1980.

[167] L. M. Schmitt, C. L. Nehaniv & R. H. Fujii. "Linear analysis of genetic algorithms". *Theoretical Computer Science*, **200**(1–2), pp. 101–134, 1998.

[168] H. Schurer, C. H. Slump & O. E. Herrmann. "Second order Volterra inverses for compensation of loudspeaker nonlinearity". In [136].

[169] G. Schwarz. "Estimating the dimension of a model". *The Annals of Statistics*, **6**(2), pp. 461–464, 1978.

[170] J. Scourias. *Overview of the Global System for Mobile Communications*. Tech. rep., Department of Computer Science, University of Waterloo, Ontario, Canada, 1996.

[171] N. Shephard. "Partial non-Gaussian state space". *Biometrika*, **81**(1), pp. 115–131, 1994.

[172] N. Shephard & M. K. Pitt. *Likelihood Analysis of Non-Gaussian Parameter-Driven Models*. Tech. rep., Nuffield College, University of Oxford, 1995.

[173] R. Shibata. "Selection of the order of an autoregressive model by Akaike's information criterion". *Biometrika*, **63**(1), pp. 117–126, 1974.

[174] T. Shiga. "Deformation distortion in disc records". *Journal of the Audio Engineering Society*, **14**(3), pp. 208–217, 1966.

[175] D. J. Spiegelhalter & A. F. M. Smith. "Bayes factors and choice criteria for linear models". *Journal of the Royal Statistical Society B*, **42**(433), pp. 213–220, 1980.

[176] J. A. Stark. *Variable Selection in Data and Signal Modelling*. Ph.D. thesis, University of Cambridge, 1995.

[177] J. A. Stark, W. J. Fitzgerald & S. B. Hladky. *Multiple-order Markov Chain Monte Carlo Sampling Methods with Application to a Change-point Model*. Tech. Rep. CUED/F-INFENG/TR.302, Department of Engineering, University of Cambridge, 1997.

[178] P. Stoica, P. Eykhoff, P. Janssen & T. Söderström. "Model-structure selection by cross-validation". *International Journal of Control*, **43**(6), pp. 1841–1878, 1986.

[179] M. A. Tanner. *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions*. Springer series in statistics. Springer-Verlag, 2nd edn., 1993.

[180] T. Teräsvirta. "Testing linearity and modelling nonlinear time series". *Kybernetica*, **30**(3), pp. 319–330, 1994.

[181] C. W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall International, 1992.

[182] L. Tierney. "Markov chains for exploring posterior distributions". *The Annals of Statistics*, **22**(4), pp. 1701–1762, 1994. With discussion.

[183] D. Tjøstheim. "Non-linear time series: A selective review". *Scandinavian Journal of Statistics*, **21**, pp. 87–130, 1994.

[184] H. Tong. *Non-Linear Time Series: A Dynamical System Approach*. Oxford Statistical Science Series. Oxford University Press, 1990.

[185] E. G. Trendell. "Tracing distortion correction". *Journal of the Audio Engineering Society*, **25**(5), pp. 273–277, 1977.

[186] P. T. Troughton. "Bayesian restoration of quantised audio signals using a sinusoidal model with autoregressive residuals". *Proceedings of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999. To appear.

[187] P. T. Troughton & S. J. Godsill. "Bayesian model selection for time series using Markov chain Monte Carlo". *Proceedings of IEEE ICASSP-97*, **V**, pp. 3733–3736, 1997.

[188] P. T. Troughton & S. J. Godsill. *A Reversible Jump Sampler for Autoregressive Time Series, Employing Full Conditionals to Achieve Efficient Model Space Moves*. Tech. Rep. CUED/F-INFENG/TR.304, Department of Engineering, University of Cambridge, 1997.

[189] P. T. Troughton & S. J. Godsill. "Bayesian model selection for linear and non-linear time series using the Gibbs sampler". In J. G. McWhirter, ed., *Mathematics in Signal Processing IV*, pp. 249–261. Oxford University Press, 1998.

[190] P. T. Troughton & S. J. Godsill. "MCMC methods for restoration of nonlinearly distorted autoregressive signals". *Proceedings of EU-SIPCO 1998*, **IV**, pp. 2029–2032, 1998.

[191] P. T. Troughton & S. J. Godsill. "Restoration of nonlinearly distorted audio using Markov chain Monte Carlo methods". *Journal of the Audio Engineering Society (Abstracts)*, **46**(6), p. 569, 1998. Preprint 4679.

[192] P. T. Troughton & S. J. Godsill. "A reversible jump sampler for autoregressive time series". *Proceedings of IEEE ICASSP-98*, **IV**, pp. 2257–2260, 1998.

[193] P. T. Troughton & S. J. Godsill. "MCMC methods for restoration of nonlinearly distorted autoregressive signals". *Signal Processing*, 1999. Submitted, awaiting review.

[194] P. T. Troughton & S. J. Godsill. "MCMC methods for restoration of quantised time series". *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, **2**, pp. 447–451, 1999.

[195] P. T. Troughton & S. J. Godsill. "Restoration of coarsely quantised audio signals using Markov chain Monte Carlo methods". Presented at the *107th Convention of the Audio Engineering Society*, 1999. To appear.

[196] M. Tsukamoto, K. Matsunaga, O. Morioka, T. Saito, T. Igarashi, H. Yazawa & Y. Takahashi. "Correction of nonlinearity errors contained in the digital audio signals". Presented at the *104th Convention of the Audio Engineering Society*, preprint 4698, 1998.

[197] S. V. Vaseghi. *Algorithms for Restoration of Archived Gramophone Recordings*. Ph.D. thesis, University of Cambridge, 1988.

[198] S. V. Vaseghi & R. Frayling-Cork. "Restoration of old gramophone recordings". *Journal of the Audio Engineering Society*, **40**(10), pp. 791–801, 1992.

[199] J. Vermaak & M. Niranjan. "Markov chain Monte Carlo methods for speech enhancement". *Proceedings of IEEE ICASSP-98*, **2**, pp. 1013–1016, 1998.

[200] J. von Neumann. "Various techniques used in connection with random digits". *National Bureau of Standards, Applied Maths Series*, **12**(3), pp. 36–38, 1951.

[201] P. J. Walmsley, S. J. Godsill & P. J. W. Rayner. "Multidimensional optimisation of harmonic signals". *Proceedings of EUSIPCO 1998*, 1998.

[202] R. A. Wannamaker. "Psychoacoustically optimal noise shaping". *Journal of the Audio Engineering Society*, **40**(7), pp. 611–620, 1992.

[203] S. Washizawa, T. Nakatani & T. Shiga. "Development of skew-sampling compensator for tracing error". *Journal of the Audio Engineering Society*, **21**(8), pp. 630–634, 1973.

[204] M. West & J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, 2nd edn., 1997.

[205] M. West, R. Prado & A. Krystal. *Latent Structure in Non-Stationary Time Series with Application in Studies of EEG Traces*. Tech. rep., Institute of Statistics & Decision Sciences, Duke University, 1997.

[206] S. A. White. "Non-linear signal processor". US Patent 4315319, 1982.

[207] S. A. White. "Restoration of nonlinearly distorted audio by histogram equalization". *Journal of the Audio Engineering Society*, **30**(11), pp. 828–832, 1982.

[208] M.-D. Wu. *Markov Chain Monte Carlo Methods Applied to Bayesian Data Analysis*. Ph.D. thesis, University of Cambridge, 1997.

[209] G. U. Yule. "On a method of investigating periodicities in disturbed series with special reference to Wolfer's sunspot numbers". *Transactions of the Royal Society of London, Series A*, **226**, pp. 267–298, 1927.

[210] P. Zhang. "Model selection via multifold cross validation". *The Annals of Statistics*, **21**(1), pp. 299–313, 1993.